Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Data-mining nad znalostmi v systému Barborka

Data-mining The Knowledge in The System Barborka

2014 **Bc. Jakub Gerlich**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jakub Gerlich**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Data-mining nad znalostmi v systému Barborka
Data-mining The Knowledge in The System Barborka

Zásady pro vypracování:

V práci nastudujte a analyzuje využití a aplikujte vybrané data-miningové metody na data získaná systémem Barborka. Hlavním cílem je poté implementace systému, která takto získané informace uživateli interpretuje ve formě umožňující další cílený rozvoj při studiu a výuce. Student se ve své práci taktéž zaměří na systém Moodle.

Zásady pro vypracování:
1. Popište jednotlivé data-miningové metody.
2. Seznamte a popište rozdílnost přístupů při využití data-miningových metod v marketingu a v e-learningu.
3. Zhodnoťte možnosti využití data-miningových metod v systému Moodle.
3. Analyzujte data získaná systémem Barborka.
5. Navrhněte strukturu uložení dat a následně použije vybrané data-miningové metody.
6. Interpretujte získané informace získaných z data-miningových metod ve formě, která umožní uživatelům další cílený rozvoj při studiu a výuce.

Seznam doporučené odborné literatury:

Ian H. Witten, Eibe Frank, Mark A. Hall: Data Mining: Practical Machine Learning Tools and Techniques,
Luboslav Lacko: Databáze: datové sklady, OLAP a dolování dat

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Marek Menšík, Ph.D.**

Datum zadání: 16.11.2012
Datum odevzdání: 07.05.2014

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *5. května 2014*

.....
podpis studenta

Poděkování

Rád bych poděkoval Ing. Radoslavu Fasugovi, Ph.D. za konzultaci k systému Barborka, jehož je autorem, Mgr. Pavle Dráždilové, Ph.D. za konzultace a Mgr. Marku Menšíkovi, Ph.D. za konzultace, rady a vstřícný přístup při vytváření této diplomové práce.

Abstrakt

Tématem této diplomové práce je data mining v oblasti vzdělávání, tedy edumining. Cílem je analyzovat, jaká data se ukládají v systémech Moodle a Barborka a posoudit možností data miningu z těchto systémů. Součástí této práce je i vytvoření data miningového modulu pro systém Barborka. V první části této práce je uvedení do problematiky data miningu a popis vybraných data miningových metod. Následuje vývoj aplikace a příklady jejího použití.

Klíčová slova

Data mining; Edumining; LMS Barborka

Abstract

The topic of this thesis is edumining which is a data mining from educational systems. The aim is to determine which data are used by educational systems Moodle and Barborka and assess the possibility of using data mining in these systems. Creation of data mining module for Barborka system is a part of this thesis. At the beginning there is the introduction to the topic and description of selected data mining methods. Several chapters are devoted to the development of the data mining module and the examples of its use.

Key words

Data mining; Edumining; LMS Barborka

Seznam použitých zkratek

Zkratka	Význam
ASP.NET	Active server pages .NET
BAL	Business access layer
DAL	Data access layer
LMS	Learning management system
MVC	Model-view-controller
VŠB-TUO	Vysoká škola báňská – Technická univerzita Ostrava

Obsah

Úvod

V současné době je běžnou praxí používat na univerzitách LMS systémy, které umožňují zjednodušení a zefektivnění výuky většího množství studentů. Tyto systémy mohou umožňovat automatizované testování studentů, distribuci výukových materiálů, nebo například konzultace s pedagogy formou diskusí. Tyto systémy evidují data, která se dají použít pro získání skrytých souvislostí a poznatků a následné zlepšení výuky.

Tato práce se zaměřuje právě na data mining z těchto LMS systémů, tedy na tzv. edumining (z angl. educational data mining). Přestože je tato práce zaměřená zejména na analýzu dat ze systému Barborka, při jejím vzniku byly vyvinuty aplikace pro analýzu dat ze systémů Barborka, Moodle a eLogika. Tyto tři systémy jsou využívány na VŠB-TUO a data miningové aplikace byly vyvinuty jako moduly třetího systému, eLogiky.

První kapitoly této práce jsou zaměřené na teoretický popis toho, co je to data mining a na aktivity, které samotnému data miningu předcházejí (Kapitola 1). Je zde porovnání využití data miningových metod v několika oblastech. Jedná se zejména o rozdíly mezi data miningem v marketingu a edumininegem. Následují informace o předzpracování dat, které je nutné pro zefektivnění data miningových metod (Kapitola 2).

V kapitole 3 jsou popsány některé data miningové metody. Jsou zde popsány zejména základní statistické metody, shlukování, rozhodovací stromy a asociační pravidla. U každé z těchto skupin jsou vybrány metody, které se buď doplňují (např. v případě shlukování se jedná o divizní přístup k-means, aglomerativní přístup simple linkage a kategoriální algoritmus quick ROCK), nebo které jsou standardně používány (např. Apriori pro tvorbu asociačních pravidel).

Po této kapitole následuje analýza systémů Barborka a Moodle (Kapitola 4). V této analýze je popsána část dat, která jsou evidována těmito systémy. V rámci této analýzy je uvedeno, jaká data se dají použít pro data mining. Je zde popsáno zejména získání dat z testů, které umožní nalezení problematických oblastí u konkrétních studentů i v celém předmětu.

Zbývající části této práce tvoří informace o tvorbě data miningové aplikace pro analýzu dat ze systému Barborka. Jsou zde popsány požadavky na tuto aplikaci (Kapitola 5), návrh (Kapitola 6) a implementace (Kapitola 7), které umožní případné navázání na tuto práci.

V kapitole 8 jsou příklady použití vytvořené aplikace k analýze záznamů, evidovaných v systému Barborka. Dále je zde ukázka práce s výslednou aplikací.

Tato práce byla podpořena z ESF projektu CZ.1.07/2.2.00/28.0209 'Elektronické opory a e-learning pro obory výpočtového a konstrukčního charakteru'.

1 Dobývání znalostí z databází

V této kapitole se budu zabývat využíváním data miningu v různých oblastech a stručným popisem toho, co se skrývá pod pojmy data mining a dobývání znalostí z databází.

Data mining je proces, který se zabývá získáváním netriviálních nebo skrytých a potencionálně užitečných informací z dat. Pod tímto pojmem se skrývá množství metod, které mohou sloužit ke klasifikaci a rozdělení dat, k získání souhrnných informací nebo k predikcím. Data mining se někdy bere jako jedna část procesu Dobývání znalostí z databází, někdy jsou však tyto výrazy považovány za synonyma.

1.1 Použití data miningových metod

Data mining je interdisciplinární pojem a je využíván v různých oblastech. Některé způsoby jeho použití jsou specifické. Obecně platí, že čím víc dat máme, tím víc se můžeme spolehnout na získané informace. Ideální ale také je, když data pokrývají co nejvíc možností a jsou co nejvíc konkrétní.

Příklad

Pokud chceme něco zjistit o obchodu ze záznamů prodeje zboží, je ideální mít tyto záznamy za několik let, abychom mohli zohlednit případné sezonní výkyvy.

1.1.1 Marketing

V oblasti marketingu se data miningové metody používají především pro analýzu klientů, hledání jejich preferencí a zlepšení nabídky produktů. Toto je možné najít v literatuře (1) (2). Toto se zajišťuje pomocí shlukové analýzy dotazníků a analýzy prodeje zboží (literatura (3)). Dále se vyhledávají frekventované kombinace zboží pro lepší rozmístění zboží v obchodě. Pomocí těchto metod je také možné predikovat množství zákazníků.

Marketing se vyznačuje zejména tím, že každý obchod denně generuje stovky až tisíce dat, která se dají pro shlukování použít. Samozřejmě je ideální, když jsou k dispozici data za co největší dobu, ale už po několika dnech shromažďování dat je možné zajistit dostatek podkladů pro některé analýzy (například pro asociační pravidla).

1.1.2 E-learning

V oblasti e-learningu je možné použít data mining, pak hovoříme o eduminingu (educational data mining) viz (4). Slouží například k nalezení skupin studentů se stejnými problémy (například v referencích (4) (5)) a k cílenému procvičování problematické látky (literatura (5) (6)). Pomocí shlukové analýzy je možné vyhledat studenty s podobnými problémy a následně jim vytvořit cvičné testy na míru, viz (6). Dále je možné predikovat úspěšnost v závislosti na předchozích výsledcích. Více v literatuře (5).

E-learning je z pohledu data miningu problematičtější, než marketing. Za prvé je v e-learningových systémech velké množství nenumerických dat (kategorie otázek, typ studia, tutor,

předchozí vzdělání,…). Za druhé je velkou nevýhodou i omezené množství dat. Jeden předmět studuje určité množství studentů a ucelené informace o nich můžeme mít po absolvování celého semestru, tedy za půl roku. Čím víc průběžných výsledků se v systému eviduje, tím je to lepší, ale ty nám mohou pomoci pouze částečně.

1.1.3 **Jiné**

Data mining se používá i v jiných oblastech, kde je potřeba zpracovávat velké množství dat. Například v medicíně se používá data mining pro analýzu DNA, jak se uvádí v článku (7). Dále se běžně používá shluková analýza na úpravu snímků z rentgenu a magnetické rezonance (reference (8)). V informatice se podobné metody používají pro analýzu obrazu, detekci hran a segmentaci (literatura (9)).

	Marketing	e-learning
Důvody	Zjednodušení nákupu zákazníkům.	Pomoc studentům při studiu, pomoc vyučujícím při plánování a přípravě studia.
Typy dat	V marketingu je dostatek numerických dat, která jsou pro data mining vhodná. Jsou zde záznamy přístupů, ceny a počty zboží v nákupech.	V e-learningu jsou kategoriální data (kategorie otázek, typ studia, tutor, předchozí vzdělání,...) i numerická data (výsledek kurzu, výsledek testu, doba vykonávání testu,...). Jsou zde informace o výsledcích a záznamy práce studenta se systémem.
Cíle	Navýšení zisku. Toto lze jednoduše objektivně vyhodnocovat pomocí tržeb a množství prodaného zboží.	Zlepšení výuky. Toto je obtížné zhodnotit, protože se jedná o subjektivní cíl.
Techniky	Tradiční data miningové metody.	Výukové systémy mají speciální vlastnosti a je potřeba využívat specifických metod, nebo upravit klasické metody. Některé je možné adaptovat, některé ne.

Tabulka 1 *Rozdíly mezi data miningem v marketingu a v e-learningu*

Výše (Tabulka Tabulka 1) je porovnání data miningu v oblasti e-learningu a marketingu. Při vytváření tohoto souhrnu se vycházelo z informací v článcích, jejichž autoři se zaměřují na data mining z e-learningových systémů (konkrétně z článků (4) a (6)) a také z vlastních zkušeností s vytvářením data miningové aplikace.

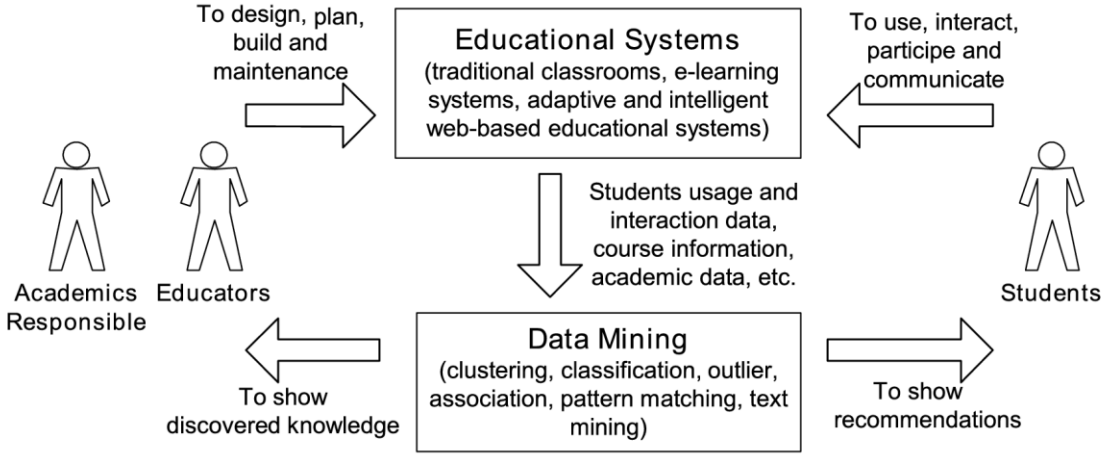
1.2 Proces dobývání znalostí z databází

Proces získávání znalostí se skládá z několika kroků. Data mining je jedním z těchto kroků. Bližší informace o dobývání znalostí z databází se nachází v literatuře (10) (11). Počet těchto kroků se různí, ale základní idea zůstává vždy stejná:

- Specifikace problému - V této části je potřeba zjistit, co chceme pomocí data miningových metod zjistit. Jestli se například jedná o vyřešení konkrétního problému, nebo o analýzu dat a získání skrytých informací.
- Získání dat - V této fázi se vyberou data, která jsou vhodná pro data mining, vyčistí a upraví se (provede se předzpracování dat). Pokud pracujeme s více zdroji dat, spojí se data z těchto zdrojů dohromady.
- Výběr metody - Uživatel musí vybrat metodu podle toho, jak chce data zpracovat.
- Předzpracování dat - Může proběhnou další předzpracování dat v závislosti na potřebách zvolené metody.
- Data mining - Samotný proces data miningu, tedy aplikace metody na data.
- Interpretace výsledků - Získání výsledků zvolené metody, jejich upravení do přijatelné formy (tabulka, graf, textový výstup) a jejich prezentace uživateli.

Obrázek Obrázek 1 ilustruje, jakým způsobem by měla probíhat interakce studentů a vyučujících s e-learningovým systémem a se systémem pro data mining.

Vyučující zadávají do systému studijní materiály a pánují termíny testování studentů. Studenti používají systém a absolvují testy. Tím se vytváří data vhodná pro data mining, která využívá data miningový systém. Tento systém má pak dvě odlišné funkce. Za prvé umožňuje studentům získat doporučení pro zlepšení jejich výsledků. Za druhé umožňuje učitelům získat skryté informace.



Obrázek 1 *Cyklus využití data miningu v e-learningových systémech (převzato z článku (4))*

Jak bylo řečeno výše, dobývání znalostí z databází se skládá z několika kroků. V následující kapitole je popsán jeden z kroků, který je potřebný pro efektivní použití data miningových metod, tedy předzpracování dat.

2 Předzpracování dat

Samotné analýze dat a použití data miningových metod předchází získání a předzpracování dat. Tato kapitola se bude věnovat pojmu preprocessing, tedy předzpracováním dat (literatura (10) (11)). Je zde popsáno, co to preprocessing je a proč je důležitý. Dále se zde budu zabývat předzpracováním dat použitým jednorázově před samotnými data miningovými metodami a předzpracováním dat používaným podle použité metody. Dále jsou zde informace o způsobu uložení těchto dat.

Předzpracování dat slouží k odfiltrování nežádoucích hodnot a k úpravě dat do vhodného formátu. Proces předzpracování dat zahrnuje několik operací, které mohou být použity pro přípravu dat do formy, která může být použita data miningovými metodami.

- Čištění - Slouží k zajištění správnosti dat. Jedná se odfiltrování nebo zpracování neúplných (například uživatel beze jména) nebo nesprávných záznamů (například vypracovaný test bez vazby na studenta). Podle toho, jak byl navržen systém, ze kterého získáváme data, může být větší či menší potřeba vyčistit záznamy od nevalidních hodnot. Například můžeme mít záznam s chybějícím atributem, který později budeme vyžadovat.
- Transformace - Tato operace už předzpracovává data, aby se zrychlily následné výpočty. Pokud víme, že budeme chtít pracovat s akademickým rokem, můžeme akademický rok vypočítat dopředu a uložit do databáze, abychom tyto výpočty nemuseli neustále opakovat. Můžeme data transformovat do námi požadovaného formátu, nebo je upravit (převody jednotek, atd.).
- Normalizace - Toto je podobné transformaci, jedná se o zjednodušení dat do základního formátu. Není nutné znát pro danou hodnotu rozsah, ve kterém se pohybuje, ale díky normalizaci máme informace o poměru dané hodnoty k maximu. Pokud například chceme normalizovat počet bodů získaný z testu, je možné nahradit počet bodů procentuální úspěšností. Tím dostaneme rovnocenné informace, protože maximum počtu bodů může být různé, ale procentuální úspěšnost je jednoznačná.
- Agregace - Spojování záznamů z několika zdrojů. Pokud je potřeba pracovat se záznamy z několika systémů najednou, je potřeba tyto záznamy spojit a sjednotit identifikátory.

Příklad

Předzpracováváme výsledky testů, které chceme importovat do datového skladu. Výsledek testu obsahuje atributy ID_testu, ID_studenta, ID_termínu čas, získané body a maximum bodů za test. Tabulka *2* obsahuje ukázku možných výsledků testů.

Záznam na prvním řádku je potřeba odstranit, protože z nějakého důvodu chybí reference na test (čištění). Nezáleží na tom, proč je záznam neúplný, je potřeba ho zpracovat. Jelikož máme dostupné údaje o testech, můžeme normalizovat bodové zisky

studentů, takže k počtu bodů můžeme přidat procentuální úspěšnost (v případě druhého záznamu je to 50%). Dále můžeme dopočítat informace, které budeme často potřebovat (transformace). Například akademický rok (2009/2010) nebo část dne, kdy probíhala aktivita (dopoledne).

ID_testu	ID_Studenta	ID_Termínu	Čas	Body	Maximum
-	12	1	12.1.2010 8:24	12	20
1	13	1	12.1.2010 8:30	10	20

Tabulka 2 *Příklad výsledků testů*

2.1 Datový sklad

Pokud chceme předzpracovaná a vyčištěná data použít víc než jednou, je potřeba uložit je do databáze. K tomuto účelu je vhodné vytvořit novou databázi, která bude sloužit výhradně účelům data miningu. Datový sklad je databáze, která je uzpůsobena pro rychlejší a jednodušší získávání velkých množství dat, viz (3) a (11). Do této databáze se přenášejí záznamy z jiné databáze, např. z té, která slouží pro každodenní práci se systémem.

V datovém skladu jsou dva typy tabulek. Jsou zde tabulky dimenzí a tabulky faktů. Dimenze obsahují data, která jsou většinou statická, může dojít k jejich změně, ale není to obvyklé. Mohou to být neměnné vlastnosti jako texty otázek a odpovědí, informace o předmětech, termíny testů, atd. Fakta jsou data, která generují uživatelé systému. Zde jsou zejména počty a délky trvání přístupů do systému, získané body, počty studujících studentů, atd.

V e-learningových systémech se eviduje velké množství informací. My jsme se zaměřili na data, která popisují chování a výsledky studentů. Proto první předzpracování dat, které je potřeba udělat, je převedení požadovaných záznamů do speciální databáze, která byla vytvořena pro potřeby data miningu.

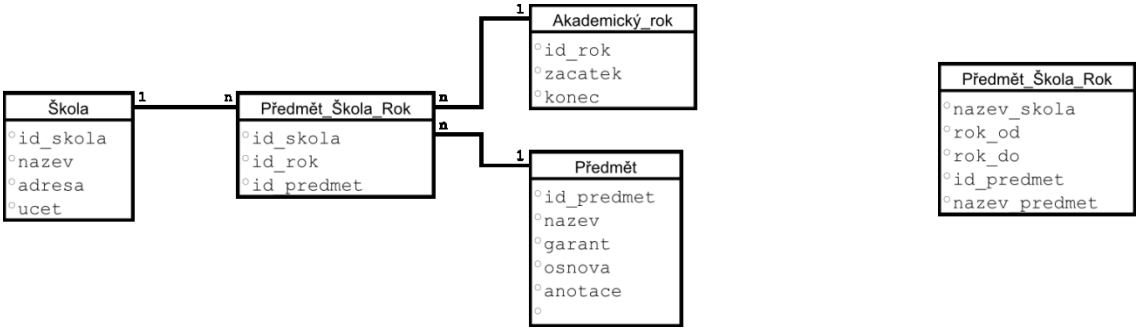
2.1.1 Výhody a nevýhody datového skladu

Jak již bylo uvedeno dříve, náš datový sklad slouží k jednoduššímu získávání dat. V datovém skladu jsou data uložena v jiné struktuře a navíc jsou zde jen vybraná data, vhodná pro edumining.

Příklad:

Mějme e-learningový systém, ve kterém jsou evidovány předměty z několika škol. Tento systém umožňuje, aby v jedné škole byl libovolný počet předmětů a aby se každý předmět mohl vyučovat ve více letech. Je jasné, že tyto informace nemohou být v jedné tabulce, ale musí být odděleny informace o škole, předmětu, akademickém roce. Rozdělením informací do více tabulek je možné vyhnout se duplicitám a umožnit jednodušší práci se systémem a editaci dat.

Když chceme přesunout data o školách a předmětech do datového skladu, můžeme si dovolit zanedbat informace, které jsou pro nás nedůležité. Můžeme vzít potřebné informace o roce, škole a předmětu a spojit je do jednoho záznamu. Tímto nám sice vzniknou duplicity, ale zbavíme se vazebních tabulek, což může urychlit přístup k některým datům. Na druhou stranu, pokud u každé školy evidujeme adresu, nějaké formální informace, kontakty a podobně, nemusíme je přesouvat do datového skladu, pokud neočekáváme, že je budeme potřebovat. Datový sklad bude sloužit pouze pro účely data miningu a proto není potřeba, aby obsahoval data, která se nepoužijí k eduminingu.



Obrázek 2 *Ukázka zanedbávání při tvorbě datového skladu. Vlevo je část ukázkové databáze a vpravo je její ekvivalent použitelný jako dimenze datového skladu.*

Pokud bychom tedy chtěli zjistit, jaké předměty jsou evidovány u nějaké školy v určitém roce, stačí nám podívat se do jediné tabulky. V původním systému bychom museli spojit 4 tabulky. Na obrázku (Obrázek Obrázek 2) je příklad nahrazení 4 tabulek jednou. Tabulka 32 obsahuje ukázku záznamů z tabulky Předmět_Škola_Rok.

nazev_skola	**rok_od**	**rok_do**	**id_predmet**	**nazev_predmet**
VŠB-TUO	2008	2009	12	Úvod do programovaní
VŠB-TUO	2009	2010	12	Úvod do programovaní
VŠB-TUO	2009	2010	13	Tvorba webových aplikací
VŠB-TUO	2011	2012	12	Úvod do programovaní
VŠB-TUO	2011	2012	14	Matematická logika

Tabulka 3 *Příklad obsahu tabulky Předmět_Škola_Rok (Obrázek Obrázek 2)*

Vše má tedy své pro a proti.

- Z příkladu je patrné, že musíme na začátku rozhodnout a zanalyzovat, jaké atributy budou použitelné pro edumining.

+ Vzhledem k tomu, že v datovém skladu úmyslně vytváříme duplicity (duplicitami v příkladu je například opakování názvu školy a předmětu ve více záznamech, viz Obrázek 2), můžeme získat některé informace jednodušeji, než v případě původní databáze.

- Jelikož se jedná o jinou databázi, musíme data přesunout a pročistit, tedy předzpracovat. Tato procedura probíhá jednou za čas, ale přesto je časově náročná. Doba trvání

záleží na tom, jaké množství dat přesouváme a jak náročné je získání a přesun námi požadovaných atributů.

+ Jelikož data přesouváme do jiné databáze, můžeme je rovnou předzpracovat. Data můžeme transformovat podle požadavků výsledného systému. Také můžeme adekvátně zpracovat neúplné nebo pochybné záznamy.

+ Při transformaci dat můžeme dopočítat nové atributy, pokud je budeme často používat.

+ Hlavní výhodou je zjednodušení práce s různými systémy. Při předzpracování dat je možné unifikovat záznamy z různých systémů a výsledná aplikace tak může být jednoduše přizpůsobena kterémukoli ze systémů Moodle, Barborka a eLogika.

Data se do datového skladu kopírují z původní databáze takzvanou datovou pumpou. Datová pumpa je tedy program, který zajišťuje export potřebných dat z původní databáze, jejich zpracování a import do nové databáze. Tento program získá data z původní databáze buď nějakým SQL dotazem, nebo zpracuje již vyexportovaný soubor dat, nebo pomocí služby. Získaná data zpracuje a naimportuje je do nové databáze. Předzpracování dat je tedy prováděno v rámci tohoto programu.

V rámci předzpracování dat je nutné čištění dat, přinejmenším ověření správnosti všech importovaných dat, aby nedošlo k chybám při pozdější práci s daty. Kromě toho se mohou některé atributy transformovat a normalizovat, aby se do databáze ukládaly ve vhodném tvaru.

2.2 Dodatečné předzpracování dat

Již dříve jsme se věnovali předzpracování dat, které je nutné pro ošetření chybných záznamů a sjednocení a přizpůsobení formátu dat. Kromě toho ale může probíhat předzpracování dat bezprostředně před použitím data miningové metody. Jedná se zejména o kategorizaci numerických dat před použitím metody, která vyžaduje na vstupu kategoriální data (například rozhodovací stromy nebo asociační pravidla), a dále je jedná o filtrování.

Kategorizace numerických nemůže být provedena při prvotním předzpracování dat, protože se jedná o ztrátovou operaci. Numerickou hodnotu, která nám dává plnou informaci a která může nabývat velkého množství hodnot, zde nahrazujeme poměrně malým množstvím kategorií. Není možné přidat další atribut, který by se jednou vypočítal, aby byla umožněna jistá variabilita. Je to z toho důvodu, aby uživatel mohl ovlivnit počet kategorií, případně aby bylo možné přizpůsobit kategorizaci právě analyzovaným datům.

Dalším předzpracováním dat mohou být úpravy a filtrování, které jsou spjaty s volbami uživatele a s parametry konkrétní data miningové metody. Například když si uživatel zvolí nějaký filtr pro analýzu určité skupiny dat. Asi nejpoužívanějším typem předzpracování dat u použité metody je právě filtrování dat, která jsou k analýze použita.

3 Vybrané metody pro dobývání znalostí z databází

Po předzpracování dat následuje samotné použití data miningových metod. V této kapitole se budu zabývat popisem vybraných statistických a data miningových metod. Někteří autoři zařazují statistiky mezi data miningové metody (jsou to například autoři publikací (4) a (6)), někteří statistiky vylučují, ale přijímají je jako nástroj pro analýzu (například (10)), ale jsou i autoři, kteří statistické metody nezmiňují vůbec (například autor publikace (12)).

V této kapitole je popsáno několik základních statistických metod. Dále jsou zde základní shlukovací algoritmy, které využívají různé přístupy k rozdělení objektů do skupin. Je zde popsán algoritmus k-means, který využívá divizní přístup ke shlukování, dále hierarchické shlukování, které funguje aglomerativně, a několik algoritmů pro práci s kategoriálními daty. V neposlední řadě jsou zde i algoritmy pro tvorbu rozhodovacích stromů a asociačních pravidel.

3.1 Základní statistiky

Tato kapitola se bude zabývat statistickými metodami, které se mohou použít pro analýzu dat z e-learningových systémů. Informace o statistice jsou v literatuře, například ve skriptech (13) (14). Jsou zde popsány základní metody, jako je aritmetický průměr, modus, medián a rozptyl.

Statistická analýza nám dokáže dát hrubou představu o datech, která jsou e-learningovým systémem evidována. Umožňuje nám oprostit se od jednotlivých záznamů a získat obecné informace. Jedná se zejména o zjištění průměrných a častých hodnot, společně s mezními hodnotami.

Statistickou analýzou můžeme zkoumat výsledky otázek, testů a kurzu.

Při analýze dat z e-learningových systémů můžeme chtít zjistit, jaké jsou průměrné výsledky různých skupin studentů. Studenty můžeme rozdělit podle akademického roku, ve kterém studují námi analyzovaný předmět, podle tutora, který vede jejich cvičení nebo i podle studijní skupiny, do které studenti náleží. Obdobně můžeme rozlišit nebo strukturovat samotné zkoumané výsledky. Máme přístup k výsledkům celého kurzu, jeho dílčích částí (např. zápočet a zkouška), jednotlivým testům a samozřejmě i k jednotlivým otázkám a odpovědím.

Díky tomu můžeme analyzovat i samotné otázky a odpovědi a zjistit tak, jestli je jejich náročnost vyhovující. Statistické metody nám například mohou pomoci odhalit otázky, které se příliš vymykají průměrům. Můžeme tak identifikovat otázky, které mají příliš nízkou, nebo naopak vysokou úspěšnost. Takovéto otázky je dobré zkontrolovat, protože je možné, že obsahují nějakou chybu. U otázek s příliš nízkou úspěšností je také možné, že jsou zadány správně, ale vyžadují zaměřit se na ně více při výuce nebo studiu.

3.1.1 Průměr

Je několik způsobů počítání průměru. Mimo jiné je možné pracovat například s aritmetickým, geometrickým nebo harmonickým průměrem. Zde je používán pouze

aritmetický průměr, který je v běžné řeči označován pouze jako průměr. Aritmetický průměr se běžně značí jako $\bar{x}$.

Aritmetický průměr nám dává základní informaci o datech, přestože bývá ovlivněn odlehlými hodnotami. Odlehlé hodnoty jsou data, která jsou výrazně vzdálená od ostatních, více ve skriptech (14). Například pokud bychom měli čísla {1, 2, 2, 2, 3, 4, 9}, pak číslo 9 je odlehlá hodnota.

Aritmetický průměr se počítá následujícím vzorcem:

$$\bar{x} = \frac{1}{n}\sum_{\{i=0\}}^{n} x_i$$

3.1.2 Modus

Modus je hodnota, která se v analyzovaném souboru dat vyskytuje nejčastěji (má nejvyšší četnost) a budeme ji značit $\hat{x}$. Na rozdíl od ostatních metod je možné při výpočtu modu pracovat i s nenumerickými hodnotami. Například modus souboru {jablko, jablko, hruška} je jablko. Jedná se tedy o typickou hodnotu.

Jeho nevýhodou je, že může výrazně kolísat, pokud se v souboru dat vyskytuje více hodnot s podobnými četnostmi. Navíc data mohou být multimodální, tedy mít víc hodnot s nejvyšší četností.

3.1.3 Medián

Medián bývá označován jako $\tilde{x}$. Tato hodnota je střed hodnot a rozděluje seřazený soubor dat na dvě stejně velké skupiny. Platí tedy, že 50% hodnot splňuje $x \leq \tilde{x}$ a zbylých 50% hodnot splňuje $x \geq \tilde{x}$.

Medián je nejméně citlivý na extrémní hodnoty a je proto vhodný pro asymetrická rozdělení, viz Tabulka 43.

3.1.4 Rozptyl

Rozptyl určuje variabilitu hodnot v souboru dat. Značí se σ^2 a je to druhá mocnina směrodatné odchylky. Dává nám představu o tom, jak rozkolísané jsou hodnoty, tedy jak blízké či vzdálené jsou od průměru. Malý rozptyl znamená, že hodnoty jsou blízké průměru, velký rozptyl naznačuje, že hodnoty mohou být vzdálené od průměru více.

Rozptyl se může počítat například následujícím vzorcem:

$$\sigma^2 = \frac{1}{n}\sum_{i=0}^{n}(x_i - \bar{x})^2$$

Pro jasnější představu o významu a změnách těchto hodnot uvádím několik příkladů, viz Tabulka 43. Na příkladech lze pozorovat, jak přínosná může být znalost rozptylu. Samotný průměr nám poskytuje pouze velmi hrubou představu o datech, ale když k němu přidáme

rozptyl, medián a modus, dozvíme se mnohem víc a můžeme si udělat přesnější představu o datech.

Příklad

Tabulka 43 obsahuje příklady vstupních dat a jejich průměrů, mediánů, modů a rozptylů. Hodnoty ve vstupních datech mohou být například zisky na burze v jednotlivých měsících. Přestože v průměru mají příklady 1, 2, 3 a 4 stejný zisk, příklad 4 obsahuje velice odlehlé hodnoty, což ukazuje rozptyl. Malý rozptyl u příkladu 3 ukazuje, že jsou vstupní hodnoty bližší průměru, než u příkladů 1 a 2.

Díky velkému rozptylu bychom i bez znalosti vstupních dat dokázali říct, že příklady 4 a 5 mají zvláštní data, která nelze uspokojivě popsat průměrem a proto je potřeba zjistit modus, případně medián.

Příklad 6 pouze ilustruje, že rozptyl nezáleží na hodnotě průměru, ale na jeho vzdálenosti od hodnot.

	vstupní data	$\bar{x}$	$\hat{x}$	$\tilde{x}$	σ^2
1	{1, 2, 3, 4, 5, 6, 7}	4	{ }	4	4
2	{2, 2, 2, 4, 6, 6, 6}	4	{2, 6}	4	3,43
3	{3, 4, 4, 4, 4, 4, 5}	4	{4}	4	0,29
4	{-62, 4, 4, 4, 4, 4, 70}	4	{4}	4	1244,57
5	{4, 4, 4, 4, 4, 4, 70}	13,43	{4}	4	533,39
6	{4, 5, 6, 7, 8, 9, 10}	7	{ }	7	4

Tabulka 4 *Ukázky základních statistik*

3.2 Shluková analýza

Další skupina metod, které se v eduminingu používají, je shluková analýza. Shlukové analýze se věnují autoři publikací (11) (15). Na začátku této podkapitoly je popsáno, co shluková analýza dělá, jaká data potřebuje a jak vypadá její výsledek. Dále se zde zabýváme některými specifiky shlukové analýzy, které u jiných data miningových metod nejsou. To se týká zejména měření vzdálenosti, respektive podobnosti objektů a měření kvality analýzy. K tomu se váže i problém s využitím různých metod podle typu dat. V neposlední řadě zde zmíníme záludnost některých shlukovacích algoritmů, které vyžadují určit počet výsledných shluků.

Ve zbytku této podkapitoly jsou pak popsány konkrétní shlukovací metody, které byly použity v rámci praktické části této práce. Jedná se o metody k-means, aglomerativní hierarchické shlukování a quick ROCK.

Shluková analýza slouží k rozdělení objektů do skupin neboli shluků. Pokud používáme pro shlukování podobnost, rozdělují se objekty tak, aby si byly objekty v rámci

jedné skupiny podobné, ale aby byly odlišné od objektů z jiných skupin. V případě vzdálenosti jsou si objekty ze stejné skupiny bližší, než objekty z různých skupin. Podobnost a vzdálenost je určena na základě hodnot atributů popisujících objekty a na zvolené metrice. Uživatel zvolí atributy objektů, které chce pro shlukování použít. Na rozdíl od statistických metod, shlukování zohledňuje několik parametrů najednou. Takže nám pomůže jednoduše oddělit například lehké a zároveň časově náročné otázky od těch lehkých, ale časově nenáročných, pokud tyto atributy použijeme k analýze (tedy úspěšnost řešení příkladu a čas, který k tomu student potřeboval). Pomocí shlukové analýzy můžeme rozdělit studenty podle jejich chování, konkrétně například podle výsledků v několika kategoriích.

Většinou je nutné upravit hodnotu atributu podle jeho významu, nebo použít správnou kombinaci atributů. Nemůžeme jako jeden atribut použít počet bodů za test a jako druhý pořadí pokusu. Body, které se pohybují v řádech desítek, by neumožnily prosazení vlivu čísla pokusu, které má hodnotu například v intervalu od 1 do 3.

Očekáváme, že nám shlukování umožní nalézt skupiny studentů, kteří mají podobné problémy. Takovýmto studentům bychom potom mohli předložit materiály pro procvičení konkrétních oblastí. K tomuto můžeme využít informace o tom, jak studenti řešili jednotlivé testové otázky.

Dále můžeme zkusit rozdělit otázky podle jejich průměrné úspěšnosti a časové náročnosti a podle získaného rozdělení upravit předpokládanou obtížnost těchto otázek. Toto předpokládá elektronické testování, díky kterému získáme informace o časové náročnosti příkladů. Navíc se můžeme pokusit shlukovat kategorie otázek podle počtů správně a špatně zodpovězených otázek a najít kategorie, které jsou nejvíc problematické.

3.2.1 Typy vstupních dat

V e-learningových systémech se vyskytují jak numerická, tak kategoriální data. Mezi numerická data patří například bodové výsledky, počty záznamů a některé časové údaje (délka trvání). Mezi kategoriální data patří například známkové ohodnocení předmětů, identifikátory objektů a některé časové údaje (datum, den v týdnu,…). Jak je popsáno v následující kapitole, práce s různými typy dat se liší. Kromě toho, že v případě numerických dat je vhodné měřit vzdálenost a pro kategoriální data se víc hodí podobnost, je vhodné rozmyslet si i použité metody.

Většina shlukovacích metod pracuje se vzdáleností a spojuje objekty, které jsou si blízké. Jedná se například o k-means a různé druhy hierarchického shlukování.

S kategoriálními daty pracují například metody CACTUS (Clustering Categorical Data Using Summaries), ROCK a Quick ROCK.

3.2.2 Měření vzdálenosti a podobnosti

Pro měření rozdílnosti objektů se používají různé metody. Podrobně jsou rozebrány například v knize (16). Buď můžeme pracovat se vzdáleností, nebo s podobností v závislosti na typu vstupních dat.

Vzdálenost vektorů x a y, pro které platí $x, y \in \mathbb{R}^n$ definujeme jako funkci $d: (\mathbb{R}^n \times \mathbb{R}^n) \rightarrow \mathbb{R}$.

Vzdálenost splňuje tyto vlastnosti (viz literatura (16)):

- Její výsledek je nezáporný: $d(x, y) \geq 0, \forall x, y \in \mathbb{R}^n$
- Vzdálenost dvou identických vektorů je nulová: $d(x, x) = 0, \forall x \in \mathbb{R}^n$
- Vzdálenost dvou vektorů je symetrická: $d(x, y) = d(y, x), \forall x, y \in \mathbb{R}^n$
- Platí trojúhelníková nerovnost: $d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in \mathbb{R}^n$

Pro tyto účely můžeme použít několik druhů počítání vzdálenosti. Nejčastěji se vzdálenost počítá následujícími způsoby:

- Euklidovská vzdálenost $x, y \in \mathbb{R}^n$:

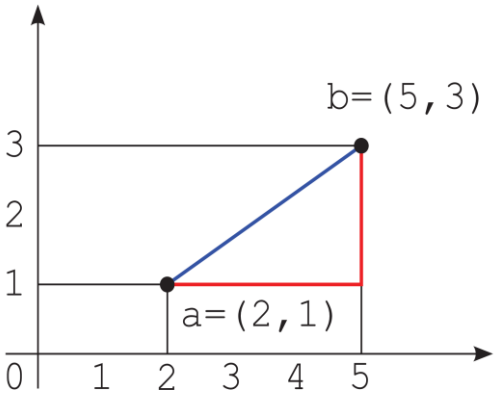
$$d(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

- Manhattanská vzdálenost $x, y \in \mathbb{R}^n$:

$$d(x, y) = \sum_{i=0}^{n} |x_i - y_i|$$

- Vážená euklidovská vzdálenost, kde w_i je váha i-tého atributu a $x, y \in \mathbb{R}^n$:

$$d(x, y) = \sqrt{\sum_{i=0}^{n} w_i (x_i - y_i)^2}$$



Obrázek 3 *Ukázka rozdílu mezi euklidovským (modrá) a manhattanským (červená) způsobem měření vzdálenosti*

Příklad

Výše (Obrázek 3) je ukázka měření vzdálenosti dvou bodů v $\mathbb{R}^2$, kde body jsou reprezentovány vektory souřadnic. Euklidovská vzdálenost bodů a a b je $d(a, b) =$

$\sqrt{(2-5)^2+(1-3)^2} = 3{,}61$. Manhattanská vzdálenost bodů a a b je $d(a,b) = |2-5|+|1-3| = 5$.

Z těchto způsobů měření vzdálenosti budeme používat zejména váženou euklidovskou vzdálenost, protože nám umožňuje nastavit významnost jednotlivých atributů a umožní tak zkombinovat jakékoli atributy. Bude však záležet na datech a nastavení vah atributů.

Jak je vidět, vzdálenost lze použít pouze v případě, že můžeme získat rozdíl atributů. Jenže v e-learningových systémech jsou i atributy, jejichž rozdíl získat nelze, nebo nedává smysl. Například identifikátor autora, označení třídy, identifikátor studenta a jiné. Nezáleží na tom, jestli jsou reprezentovány číslem nebo řetězcem. S těmito kategoriálními daty musíme také nějak počítat.

Při měření vzdálenosti kategoriálních atributů si můžeme pomoct tím, že určíme dvě konstanty - jednu pro identické hodnoty (zde by měla být nula, abychom zachovali pravidlo nulové vzdálenosti identických objektů) a jednu pro rozdílné hodnoty. Pokud tedy budou mít dva objekty x a y kategoriální atribut i stejný, bude platit $x_i - y_i = 0$. Jestliže se v tomto atributu liší, platí $x_i - y_i = 1$.

Jiná možnost, jak pracovat s kategoriálními atributy, je měření podobnosti. Podobnost můžeme měřit například pomocí Jaccardova indexu (také známý jako Jaccardův podobnostní koeficient), více v literatuře (16). Jeho vzorec pro $X, Y \subset K^{\mathrm{n}}$ (kde K je množina kategorií všech atributů) je velmi jednoduchý:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Jaccardův index měří podobnost dvou množin takovým způsobem, že porovná společné prvky a všechny prvky, které se v nich vyskytují. Z toho důvodu musíme objekty (vektory) reprezentovat množinami.

Příklad

Mějme objekty s atributy $\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \mathrm{a}_4$ a konkrétní vektory hodnot těchto atributů: $c = (1, A, 1, jablko)$ a $d = (1, C, 2, jablko)$. Tyto vektory nahradíme množinami tak, že každý prvek množiny bude reprezentovat atribut a jeho hodnotu. Vzniknou následující množiny, které nahradí původní vektory: $C = \{a_1 1,\ a_2 A,\ a_3 1, a_4 jablko\}$ a $D = \{a_1 1,\ a_2 C,\ a_3 2,\ a_4 jablko\}$. Abychom spočítali Jaccardův index, musíme zjistit průnik a sjednocení těchto dvou množin. Průnik obsahuje první a poslední prvek každého vektoru, protože ty jsou identické: $C \cap D = \{a_1 1, a_4 jablko\}$. Sjednocení obsahuje kromě prvků v průniku i prvky, které jsou rozdílné: $C \cup D = \{a_1 1, a_2 A, a_2 C, a_3 1, a_3 2, a_4 jablko\}$. Jaccardův index má tedy hodnotu

$$J(C,D) = \frac{|\{a_1 1, a_4 jablko\}|}{|\{a_1 1, a_2 A, a_2 C, a_3 1, a_3 2, a_4 jablko\}|} = \frac{2}{6} = 0{,}33$$

Jaccardův index se tedy hodí převážně pro podobnost kategoriálních dat, jelikož nezohledňuje, jak jsou si hodnoty podobné. Určuje pouze to, jestli jsou dvě hodnoty atributu stejné, nebo ne.

Jak bylo dříve řečeno, můžeme podobnost, změřenou pomocí Jaccardova indexu, přepočítat na vzdálenost. Každopádně, vždy když budeme pracovat s numerickými i kategoriálními daty dohromady, musíme rozhodnout, jestli budeme počítat vzdálenost a pro kategoriální data použít konstanty, nebo počítat podobnost a numerická data určitým způsobem rozdělit do kategorií.

3.2.3 Měření kvality shlukování

Různými shlukovacími algoritmy můžeme počáteční množinu objektů rozdělit různými způsoby. Musíme být tedy schopni říct, který způsob a které rozdělení je lepší. Pro tyto účely existuje několik způsobů, jak lze ohodnotit kvalitu shlukování. Kvalitu můžeme měřit pomocí různých kritérií:

- Vzdálenost objektů uvnitř shluku (můžeme pracovat se vzdáleností nejbližšího nebo nejvzdálenějšího objektu od středu shluku, do kterého patří, případně s průměrnou vzdáleností všech prvků shluku).
- Vzdálenost shluků (můžeme porovnávat vzdálenosti středů shluků nebo nejbližších prvků nepatřících do stejného shluku).
- Rovnoměrnost rozložení objektů uvnitř shluku (například jak se liší vzdálenost jednotlivých objektů od středu od průměrné vzdálenosti objektu od středu).
- Rovnoměrnost rozložení do shluků (například porovnávání počtu objektů v jednotlivých shlucích).
- Výskyt hodnot konkrétního atributu v daném shluku a v ostatních shlucích (přestože shlukování zohledňuje všechny analyzované atributy najednou, může být měřítkem kvality porovnání výskytů jednotlivých hodnot v různých shlucích, tedy například entropie, více o entropii v části o rozhodovacích stromech).

Dva z možných způsobů měření kvality shlukování jsou popsány níže:

Davies–Bouldin index

Davies-Bouldin index lze vypočítat následujícím vzorcem, kde n je počet shluků, σ_x je průměrná vzdálenost prvků shluku x od jeho středu c_x a $d(c_x, c_y)$ je vzdálenost středů shluků x a y.

$$I = \frac{1}{n}\sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{\mathrm{j}=1}^{\mathrm{n}}\max_{\mathrm{i}\neq\mathrm{j}}\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}$$

Čím má naše rozdělení tento index bližší nule, tím je rozdělení lepší. Znamená to, že průměrná vzdálenost mezi objekty v jednom shluku je nízká a naopak vzdálenost mezi středy různých shluků je velká.

Dunn index

Dunn index lze vypočítat následujícím vzorcem, kde n je počet shluků, $d'(c_x)$ je vzdálenost mezi dvěma nejvzdálenějšími objekty ve shluku x a $d(c_x, c_y)$ je vzdálenost středů shluků x a y.

$$I = \min_{1 \le i \le n} \left(\min_{1 \le j \le n, i \ge j} \left(\frac{d(c_i, c_j)}{\max_{1 \le k \le n} (d'(k))} \right) \right)$$

Tento index se chová trochu jinak, než předcházející. Za prvé, čím je vzdálenost středů shluků větší, tím má i index větší hodnotu, takže čím větší hodnota indexu, tím lepší je rozdělení. Za druhé, tím, že pracujeme s maximy a minimy, získáme pouze informaci o nejhorším rozdělení, tedy poměr nejbližších shluků a největší vzdálenosti uvnitř největšího shluku.

3.2.4 Problémy s určením parametrů analýzy

Jak bylo popsáno dříve, shluková analýza rozděluje záznamy do skupin (shluků). Po zvolení konkrétní shlukovací metody, případně algoritmu je většinou nutné zadat ještě parametry shlukování. Většinou se zadává počet výsledných shluků.

Nevyhneme se tomu, aby uživatel zadával počet výsledných shluků, nebo nějaký jiný parametr, který taky nebude lehké zvolit. Metoda k-means se bez počtu shluků (ono k v názvu metody) prostě neobejde a potřebuje ho již od začátku. Oproti tomu, při hierarchickém shlukování, potřebujeme znát počet shluků pouze k tomu, abychom věděli, kdy má shlukování skončit. Naopak quick ROCK potřebuje místo počtu shluků threshold, tedy hraniční podobnost, kdy jsou si dva záznamy ještě podobné.

Výsledky shlukovacích metod jsou vždy velmi závislé na volbě těchto parametrů (počet shluků nebo threshold). V závislosti na datech může například u algoritmu quick ROCK drobná změna thresholdu zapříčinit to, že místo desítek shluků vznikne pouze jediný. Podobně u metody k-means může nevhodný počet shluků způsobit naprosto nevyhovující rozdělení.

Je možné uživateli trochu pomoci tím, že se vypočítá několik rozdělení shluků v blízkém okolí zadaného parametru a vybere se to nejkvalitnější. Ale i s touto pomůckou (která samozřejmě zvyšuje časovou náročnost analýzy) je potřeba promyslet a vyzkoušet, jaký počet shluků zadat.

3.2.5 K-means

Metoda k-means rozdělí objekty do zadaného počtu shluků (k) tak, aby se minimalizovala vzdálenost každého objektu od středu jeho shluku, více v literatuře (15). Existují různé varianty.

Jedním dělením je dělení podle středu shluku. Středem shluku může být střední hodnota (k-means). Dalším středem může být nejčastější hodnota, tedy medián (k-medians). U obou

těchto variant můžeme počítat buď se samotným středem, nebo s objektem, který je mu nejblíže. Díky této úpravě nám extrémně vzdálený objekt moc nenaruší shluky.

K-means s opakovaným půlením je varianta, při které na začátku všechny objekty patří do jednoho shluku, který se pomocí k-means (nebo nějaké jeho varianty) rozdělí na dva shluky. Z nich se jeden vybere (podle velikosti, kvality shluku, rozložení objektů uvnitř shluku, …) a ten se znovu rozpůlí. Takto se postupuje až do chvíle, kdy je dosaženo požadovaného počtu shluků.

Fuzzy shlukování nepočítá s jednoznačným rozdělením do disjunktních shluků. Každý objekt má pro každý shluk určenu pravděpodobnost, se kterou náleží do daného shluku. Tím dokážeme lépe popsat rozložení objektů ve shlucích a můžeme také identifikovat objekty, u kterých nelze jednoznačně určit, do kterého shluku patří.

- Vstup: Vstupem shlukovací metody k-means je přirozené číslo *k* určující výsledný počet shluků a pole objektů (vektorů), které se mají do shluků rozdělit. Kromě toho je možné určit, jakým způsobem se bude měřit vzdálenost (Euklid nebo Manhattan).
- Výstup: Výstupem je *k* shluků. Každý shluk má reprezentanta (střed) a pole objektů (vektorů), které do něj patří.
- Algoritmus: Zjednodušený kód algoritmu je umístěn níže. Na začátku tento algoritmus náhodně určí středy shluků (metodou GenerateMeans). Zbytek se opakuje v cyklu, dokud jsou prováděny nějaké změny. Ke každému objektu ze vstupu se najde shluk, do kterého by měl patřit, tedy takový, od jehož středu je nejméně vzdálený (metodou FindCluster). Následně se přepočítají středy shluků (metoda RecalculateMeans}vrací 1, pokud nastala změna a 0, pokud se nezměnil střed žádného shluku) a pokud nastala nějaká změna, cyklus se opakuje.

```
List<Cluster> CreateClusters(int k, List<Vector> vectors)
{
      List<Cluster> clusters = GenerateMeans(k);
      bool changed = true;
      while (changed)
      {
            foreach (Vector v in vectors)
            {
                  Cluster nearest = FindCluster(clusters,v);
                  nearest.Objects.Add(v);
            }
            changed = RecalculateMeans(clusters);
      }
      return clusters;
}
```

Kód 1 *Algoritmus k-means*

Aglomerativní hierarchické shlukování

Hierarchické shlukování je typ shlukování, při kterém při tvorbě výsledných shluků dochází k dělení (divizní) nebo spojování (aglomerativní) existujících shluků. U divizního shlukování vycházíme z jednoho shluku, který obsahuje všechny objekty, a tento postupným

dělením rozdělíme do požadovaného počtu shluků. Příkladem tohoto typu shlukování je i k-means s postupným půlením.

Aglomerativní hierarchické shlukování postupuje přesně opačně. Začínáme s tolika shluky, kolik je objektů, a tyto postupně spojujeme do větších shluků. Podle toho, jakým způsobem vybíráme shluky pro spojení, existuje několik variant této metody. Můžeme spojovat shluky s nejbližším středem, nebo třeba takové shluky, jejichž nejbližší (nebo naopak nejvzdálenější objekty) mají nejmenší vzdálenost.

Vstupem těchto algoritmů bude zase počet shluků k, při jehož dosažení má algoritmus skončit, a pole objektů, které chceme shlukovat. V takovém případě by bylo výsledkem zase k shluků.

Můžeme ale jako vstup použít pouze pole shlukovaných objektů a provést algoritmus až do konce (tedy do stavu, kdy jsou všechny objekty spojeny do jednoho shluku). Výstupem by v tomto případě byl strom. Každý uzel by představoval jeden shluk a listy by byly shlukované objekty. V tomto případě bychom názorně viděli hierarchii a postup shlukování.

Algoritmus simple linkage je jedním z aglomerativních shlukovacích algoritmů. Pracuje tak, že najde dva objekty (z různých shluků), které jsou nejméně vzdálené a sloučí shluky, do kterých tyto objekty náleží. Takto postupuje, až zůstane pouze zadaný počet shluků.

- Vstup: Vstupem shlukovací metody simple linkage je přirozené číslo k určující výsledný počet shluků a pole objektů (vektorů), které se mají do shluků rozdělit. Kromě toho je možné určit, jakým způsobem se bude měřit vzdálenost (Euklid nebo Manhattan).
- Výstup: Výstupem je k shluků. Každý shluk má reprezentanta (střed) a pole objektů (vektorů), které do něj patří.
- Algoritmus: Zjednodušený kód algoritmu je umístěn níže}. Na začátku tento algoritmus vytvoří tolik shluků, kolik je vstupních objektů (metodou CreateClusters). Zbytek se opakuje v cyklu, dokud je počet shluků větší, než požadované k. Najdou se shluky, jejichž objekty jsou si nejbližší a tyto shluky se sloučí do jednoho. Dva shluky jsou si nejbližší, pokud každý z nich obsahuje jeden z dvojice nejbližších objektů. Dva objekty jsou si nejbližší, pokud jsou v různých shlucích a jejich vzdálenost je nejmenší ze vzdáleností všech dvojic objektů z různých shluků.

```
List<Cluster> CreateClusters(int k, List<Vector> vectors)
{
      List<Cluster> clusters = CreateClusters(vestors);
      while (clusters.Count>k)
      {
            Cluster cluster1, cluster2;
            FindNearest(vectors, out cluster1, out cluster2);
            Join(cluster1, cluster2);
      }
      return clusters;
}
```

Kód 2 *Algoritmus simple linkage*

ROCK

ROCK je shlukovací algoritmus určený přímo pro kategoriální data (literatura (17)). Tento algoritmus porovnává podobnost objektů. Ta se může počítat například pomocí Jaccardova indexu. Pokud jsou si dva objekty podobnější, než uživatelem zadaná hraniční podobnost θ, vytvoří mezi nimi spojení. Následně vytváří shluky tak, aby bylo co nejméně spojení mezi jednotlivými shluky a co nejvíce spojení uvnitř shluků. Uživatelem zadané θ je používáno i v kriteriálních funkcích, podle kterých se určuje, zda se mají dva shluky spojit nebo ne.

3.2.6 **Quick ROCK**

ROCK by měl být efektivní a pro kategoriální data je velmi vhodný. My jsme se však rozhodli pro algoritmus QuickROCK, který je popsán v literatuře (18). QuickROCK je zjednodušená a zrychlená verze algoritmu ROCK. Tento algoritmus také porovnává podobnost každých dvou objektů s uživatelem zadanou hraniční podobností θ, ale shluky vytváří tak, aby mezi shluky nebylo žádné spojení.

Tento algoritmus je nesrovnatelně rychlejší a také z uživatelského hlediska je výhodnější. I zde musí uživatel zadat hraniční podobnost θ, která rozhodne o tom, jestli budou dva objekty spojeny nebo ne. Ale v tomto případě je jen omezené množství hodnot, kterých může θ nabývat. Jsou to právě hodnoty Jaccardova indexu pro danou velikost vektoru.

Příklad: Pokud u objektů porovnáváme 5 atributů, může θ nabývat právě šesti hodnot podle toho, kolik hodnot mohou mít dva vektory společných (0/10, 1/9, 2/8, 3/7, 4/6, 5/5). To uživateli výrazně zjednoduší výběr, přestože výsledky budou trochu jiné, než u algoritmu ROCK.

Vstupem je tedy pole objektů pro shlukování a hraniční podobnost θ, která nabývá hodnot mezi 0 a 1. Výstupem bude jeden nebo více shluků, podle toho, jaké θ uživatel zadal a jaká jsou data.

Příliš nízká hodnota θ způsobí, že vznikne pouze jeden shluk, protože bude mezi objekty příliš mnoho vazeb.

Naopak příliš vysoká hodnota (např. $\theta = 1$) způsobí, že v každém shluku budou objekty, které mají identické atributy. To se hodí pro předzpracování záznamů pro tvorbu asociačních pravidel. Například místo 100 identických záznamů nám vznikne jeden shluk, který reprezentuje těchto 100 záznamů, které se ničím neliší. Takovýmto předzpracováním nic neztratíme a pouze urychlíme následné analýzy.

- Vstup: Vstupem shlukovací metody quick ROCK je hraniční podobnost θ, která je v intervalu < 0;1> a pole objektů (vektorů), které se mají do shluků rozdělit.
- Výstup: Výstupem je pole shluků. Každý shluk má pole objektů (vektorů), které do něj patří.
- Algoritmus: Zjednodušený kód algoritmu quick ROCK je umístěn níže. Algoritmus projde všechny vstupní vektory a u každého najde shluky, kterým je blízký (metoda

FindSimilarClusters najde shluky, které obsahují alespoň jeden objekt, který je danému vstupnímu vektoru podobnější, než zvolené θ).

Podle toho, kolik shluků je nalezeno, se liší další postup. Pokud není nalezen žádný dostatečně podobný shluk, je vytvořen nový a tento vektor je k němu přidán. Pokud je nalezen právě jeden shluk, vektor se k němu pouze přidá. Pokud je nalezeno větší množství blízkých shluků, tyto shluky se spojí do jednoho (metoda Join) a samozřejmě je do něj přidán i zkoumaný vektor.

```
List<Cluster> CreateClusters(double theta, List<Vector> vectors)
{
      List<Cluster> clusters;
      List<Cluster> similar;
      foreach (Vector v in vectors)
      {
            similar = FindSimilarClusters(clusters, v, theta)
            if (similar.Count==0)
                  clusters.Add(new Cluster(v));
            else if (similar.Count==1)
                  similar[0].Add(v);
            else
            {
                  similar[0].Add(v);
                  Join(similar);
            }
      }
      return clusters;
}
```

Kód 3 *Algoritmus quickROCK*

3.3 Rozhodovací stromy

Rozhodovací stromy také umožňují rozdělení objektů podle uživatelem vybraných vlastností, ale objekty rozdělují v jedné chvíli podle jedné vlastnosti. Informace o těchto metodách jsou v literatuře (19) (15). Navíc jsou objekty rozdělovány takovým způsobem, aby byly výsledné skupiny homogenní vzhledem ke zvolené „predikované“ vlastnosti. Kromě výsledných skupin nám rozhodovací stromy umožňují vidět i způsob rozdělení objektů a kromě analýzy existujících záznamů umožňují s určitou přesností „predikovat“ a hledat vztahy a korelace mezi daty. Pomocí rozhodovacích stromů můžeme zase rozdělit otázky do skupin podle obtížnosti, můžeme rozdělit studenty do skupin podle úspěšnosti ve zvládání testů nebo kurzu, nebo můžeme rozdělit kapitoly podle obtížnosti.

Rozhodovacími stromy můžeme zkusit rozdělit otázky například podle obtížnosti, času potřebného k jejich zodpovězení a autora a získat tak podrobnější informace, než pomocí shlukové analýzy, například jaký je poměr mezi lehkými a těžkými otázkami určitého autora. Na druhou stranu, při použití shlukování se můžeme jednodušeji dostat k analyzování skupin otázek, které jsou největší.

Dále můžeme zkusit analyzovat termíny nebo testy a získat tak například informaci o tom, jestli se v dopoledních termínech vyskytuje víc lepších výsledků, než v termínech odpoledních.

3.3.1 **ID3**

Jedním z algoritmů pro výpočet rozhodovacích stromů je ID3, více v referencích (20). Tento algoritmus přijme na vstupu pole objektů (vektorů), které bude analyzovat a také atribut, který má strom predikovat. Výstupem je co nejmenší strom, který vypadá následovně. Kořen obsahuje všechny objekty a každý uzel rozděluje objekty podle hodnot jednoho atributu. Listy obsahují objekty, které mají stejnou hodnotu predikovaného atributu.

Atribut, podle kterého se záznamy v daném uzlu rozdělí, je určen pomocí entropie. Entropie S značí míru neuspořádanosti. Výpočet je následující:

$$S = -\sum_{\mathrm{i}}^{\mathrm{n}} \frac{P_i}{\ln P_i}$$

P_i je pravděpodobnost, že objekt ve zkoumané množině objektů má hodnotu predikovaného atributu rovnu i.

Výhoda rozhodovacích stromů je ta, že jsou přehledné a snadno interpretovatelné. Čím blíže je uzel stromu ke kořeni, tím větší má daný atribut význam pro rozdělení výsledků. Navíc, čím kratší je větev, tím méně atributů má vliv na predikovaný atribut.

- Vstup: Vstupem algoritmu ID3 je pole objektů (vektorů), pole atributů a atribut, který má být stromem predikovaný.
- Výstup: Výstupem je stromová struktura, jejíž listy jsou homogenní vzhledem k predikovanému atributu.
- Algoritmus: Zjednodušený kód algoritmu ID3 je umístěn níž. Algoritmus zkontroluje, zda dostal validní data, jinak vrátí prázdný strom. Potom zkontroluje, jestli už data nejsou homogenní vzhledem k predikovanému atributu (metoda IsHomogeneous). Pokud ano, vrátí strom s jedním uzlem, protože víc jich už není potřeba. Pokud ne, pomocí entropie najde atribut, který data nejlépe rozděluje (metoda BestAttribute a vytvoří jednu větev pro každou hodnotu, které tento atribut může nabývat.

```
Tree ID3(List<Vector> vectors, Attribute targetAttribute,
List<Attribute> attributes)
{
    Tree root = new Tree();
    if (vectors is null)
        return root;
    root = new Tree(vectors);
    if (IsHomogeneous(vectors,targetAttribute))
        return root;
    if (attributes is null)
        return root;
```

```
    else
    {
        Attribute a = BestAttribute(vectors,targetAttribute);
        foreach (var value of a)
        {
            root.AddBranch(ID3(vectors.Where(a=value),
targetAttribute, attributes.Remove(a)))
        }
    }
}
```

Kód 4 *Algoritmus ID3*

Asociační pravidla

Asociační pravidla se hodí například k predikcím výsledků testů, jelikož se pokoušejí najít skryté vztahy. Informace o těchto metodách jsou v literatuře (20) (15). Zde využijeme Apriori algoritmu, který vyhledává frekventované kombinace. Právě díky jejich nalezení můžeme vytvořit asociační pravidla. Když máme nalezeny kombinace, můžeme vybrat takové, které mají dostatečný počet výskytů (podporu). U těch můžeme zjistit jejich podporu a spolehlivost:

$$Podpora(X \rightarrow Y) = P(X \cup Y)$$

$$Spolehlivost(X \rightarrow Y) = P(Y|X) = \frac{|X \cap Y|}{|X|}$$

X a Y jsou hodnoty atributů. Podpora pravidla $X \rightarrow Y$ je určena tím, jaké procento záznamů obsahuje $X \rightarrow Y$ Spolehlivost pravidla $X \rightarrow Y$ je určena podmíněnou pravděpodobností. Je to tedy procento záznamů obsahujících X, které obsahují i Y

Příklad

Mějme databázi 500 studentů a jejich výsledků z různých předmětů. Pokud budeme tyto záznamy zkoumat, můžeme zjistit, že 100 studentů mělo výslednou známku jedna z matematiky i z logiky (takováto kombinace má podporu 20%). Pak už stačí zjistit, jakou spolehlivost budou mít obě varianty, tedy jestli dobrý výsledek z matematiky může znamenat dobrý výsledek z logiky, nebo jestli naopak dobrý výsledek z logiky může znamenat dobrý výsledek z matematiky. Nebo můžeme zjistit, že přestože se tato kombinace vyskytuje dostatečně často, spolehlivost je nízká a není tam tedy žádný vztah.

Při analýze e-learningových systémů můžeme hledat vztahy mezi různými výsledky nebo mezi výsledkem a vlastnostmi studenta, testu termínu, atd.

3.3.2 Volba thresholdu

Při tvorbě asociačních pravidel je potřeba určit dva parametry: hraniční podporu a hraniční spolehlivost. Volba těchto hodnot výrazným způsobem ovlivňuje dobu výpočtu pravidel, jejich množství a samozřejmě relevanci.

Podpora udává, kolik procent vstupních objektů musí obsahovat konkrétní kombinaci hodnot atributů. Čím větší je tento parametr, tím rychleji trvá výpočet pravidel a tím méně pravidel získáme. Pokud pracujeme s velkými objemy dat, není potřeba nastavovat vysokou podporu, protože například podpora 10% je pro 1000 záznamů dostatečná, ale pro 50 záznamů je to už málo. Záleží na uživateli, kde si určí hranici.

Spolehlivost už neovlivňuje rychlost výpočtu, protože se používá až po vygenerování pravidel. Zase ale platí, že při zvolení příliš nízké spolehlivosti budeme zahlceni množstvím vygenerovaných pravidel. Spolehlivost by měla zůstat přiměřeně velká, ať už pracujeme s desítkami záznamů, nebo s tisíci.

3.3.3 **Apriori**

Informace o algoritmu jsou v literatuře (15) (21). Vstupem pro asociační pravidla je pole objektů (vektorů) a hraniční hodnoty podpory a spolehlivosti. První se pomocí algoritmu Apriori vyhledají všechny frekventované kombinace atributů, které mají větší podporu, než hranice zadaná uživatelem. Následně se z těchto frekventovaných kombinací vytvoří pravidla a vyberou se ta, která mají větší spolehlivost, než hranice zadaná uživatelem.

Pomocí asociačních pravidel můžeme zjistit, jestli může být vztah mezi získanými body (nebo dobou trvání testu) a některými volitelnými parametry. Testy jsou organizovány v různé dny v týdnu a různé časy, takže s dostatečným množstvím dat je možné ptát se, jestli může být výsledek ovlivněn dnem v týdnu nebo hodinou konání testu. Také můžeme zjistit počet studentů na termínu a můžeme hledat vztahy mezi naplněním termínu nebo formou studia a výsledkem.

V neposlední řadě můžeme zkusit zjistit, jestli s rostoucím počtem pokusů o zvládnutí testu roste úspěšnost. Pokud jsou studenti, kteří se o zvládnutí testu pokoušeli třikrát a vícekrát, můžeme zkoumat, jestli se jejich výsledky postupně zlepšovaly.

- Vstup: Vstupem algoritmu Apriori je pole objektů (vektorů), pole antecedentů, pole konsekventů, minimální podpora a minimální spolehlivost.
- Výstup: Výstupem je pole pravidel ve tvaru $X \rightarrow Y$, kde u každého pravidla je ještě informace o podpoře a spolehlivosti daného pravidla.
- Algoritmus: Algoritmus pomocí algoritmu Apriori najde frekventované kombinace hodnot atributů se zadanou minimální podporou. Následně projde tyto kombinace a vybere takové, které mají dostatečnou spolehlivost a správné atributy na pozicích antecedentů a konsekventů.

4 Analýzy vybraných e-learningových systémů

Tato kapitola je zaměřená na analýzu vybraných e-learningových systémů, u kterých bylo potřeba zjistit, jaké data evidují a jaká data je možné použít pro účely data miningu. Jedná se hlavně o systém Barborka, který byl vyvíjen a používán na VŠB-TUO. Dalším systémem byl Moodle, což je otevřený systém pod obecnou veřejnou licencí GNU, který je využíván na různých školách.

4.1 Analýza systému Barborka

LMS Barborka je e-learningový systém pro přípravu výukových materiálů, výuku a testování znalostí. Vývoj současné verze začal v roce 2001 na FEI VŠB-TUO, přestože první verze tohoto systému začala vznikat už v roce 1982. Následující část je zaměřena na data z tetování a záznamy přístupů do systému, které jsou použitelné pro účely data miningu.

4.1.1 Testy

Pro samotný test je stěžejní tabulka *test*, kde jsou uloženy informace o testech, jako například název a popis testu, počet otázek a kurz, jehož je test součástí. U každého testu je možné upravovat jeho strukturu v tabulce *test_struct*, což znamená, že je možné, respektive nutné, určit kapitoly, ze kterých se mají vybírat jednotlivé testové otázky. Kapitoly jsou jemnějším dělením kurzů, s každou kapitolou může být spojeno několik otázek a stejně tak jedna otázka může být připojena k několika kapitolám.

Uživatel, který tvoří test, může určit, kolik otázek se má vybrat ze které kapitoly a jaké má být jejich bodové ohodnocení. Díky tomu mohou být testy generovány automaticky (pseudonáhodně) a přitom mít určitou strukturu a dělení.

Zároveň je ještě možné zúžit výběr pouze na několik otázek z dané kapitoly. To je možné přidáním záznamu do tabulky *test_question_detail*. Tímto způsobem lze například zařídit, aby se v každém automaticky vygenerovaném testu objevila stejná otázka.

Všechny otázky jsou uloženy v tabulce *question*, která obsahuje informace o otázkách, jako je kurz, u kterého je otázka využita, samotný text otázky, čas na otázku, datum poslední změny a další informace. Atributem, který nás asi zajímá nejvíce, je typ otázky. Základní typy jsou variantní a tvořená.

- Variantní otázka – správnou odpověď student vybírá z několika variant, kde určitý počet odpovědí je správný a zbytek je špatný.
- Tvořená otázka – student musí správnou odpověď sám vymyslet a zapsat. U tohoto typu otázek je možné určit formát odpovědi (číslo, řetězec, vektor, …)

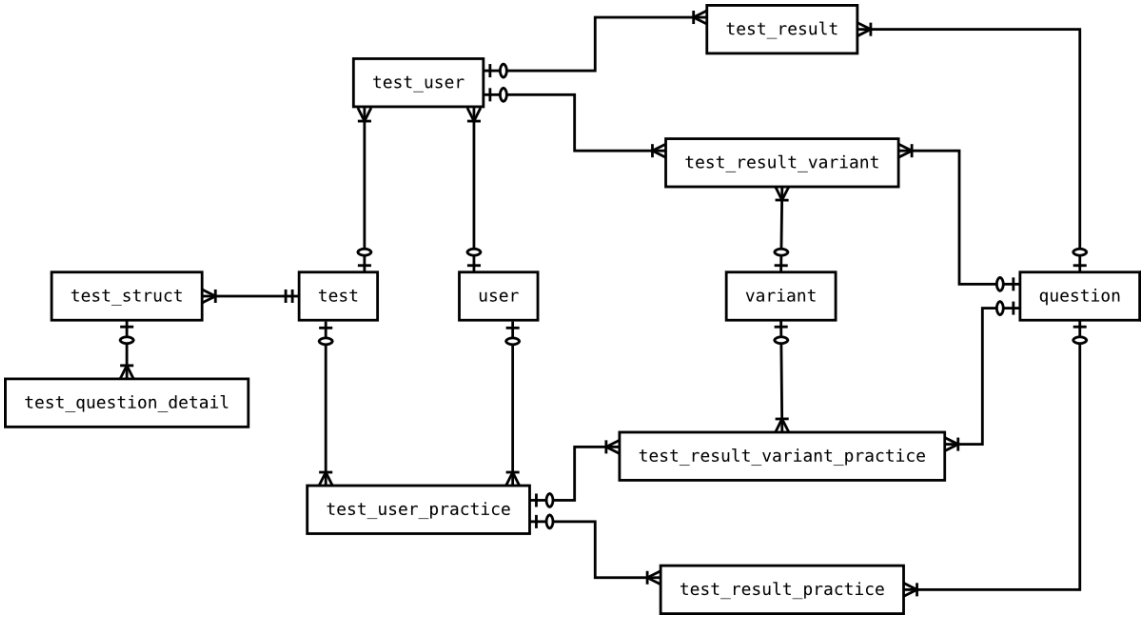
4.1.2 Výsledky testů

Výsledky testů jsou rozděleny na dvě části. Jednu tvoří výsledky cvičných testů a druhou výsledky hodnocených testů. Z bezpečnostních důvodů byly tyto dvě skupiny testů

odděleny a tak jsou všechny tabulky spojené s vyhodnocováním testů v databázi přítomny dvakrát. Jedna verze těchto tabulek má sufix *_practice*. Já na následujících řádcích popíšu pouze tabulky hodnocené, přičemž tabulky pro cvičné testy jsou naprosto identické a plní stejnou funkci. V tabulce *test_user* jsou výsledky všech testů, které uživatel vykonal. Záznamy v této tabulce obsahují informaci o začátku a konci testu, bodové ohodnocení, výsledek testu a IP adresu počítače, na kterém byl test řešen. Dalšími atributy jsou samozřejmě identifikátory testu a varianty daného testu, testová skupina a identifikátor uživatele. V této tabulce jsou tedy konečné výsledky testů. Výsledky jednotlivých otázek jsou v tabulce *test_result*.

Tabulka *test_result* obsahuje záznamy o výsledcích jednotlivých otázek. V této tabulce jsou záznamy spojeny pouze s otázkou a s konkrétní variantou testu, už zde není informace o uživateli, který test vypracovával (tato informace se však dá zjistit z tabulky *test_user*). Další atributy záznamů jsou maximální počet bodů za danou otázku, získaný počet bodů a pořadí otázky. Pokud uživatel musí odpověď na otázku sám zadat, je zde i tato odpověď.

V případě, že je otázka variantní, nachází se odpovědi v tabulce *test_result_variant*, která obsahuje varianty variantních otázek testů. Záznamy jsou tedy spojeny s otázkou, odpovědí a s konkrétní variantou testu.



Obrázek 4 *ER diagram části databáze systému Barborka, která je spjatá s testy a jejich výsledky*

4.1.3 **Záznamy přístupů**

Další částí systému, která je vhodná pro data mining jsou záznamy přístupů na stránky. Zvláště u e-learningových systémů je žádaná informace o tom, ke kterým stránkám uživatelé přistupují a jak dlouho na nich jsou.

Informace o době zobrazení dané stránky nemá hned z několika důvodů velkou vypovídací hodnotu. Přesto pomocí ní alespoň můžeme určit, jestli se na danou stránku uživatel dostal omylem, nebo ji využil jen k přístupu k jiné stránce, nebo byla jeho cílem.

V tabulce *inspectlog* jsou záznamy o přístupech ke stránkám. Tyto záznamy jsou celkem obsáhlé. Obsahují informace jako IP adresu uživatele, zadanou URL adresu a soubor, který byl zobrazen, odpovídající subsystém a jestli byla zaznamenána nějaká chyba. Dále obsahují přesný čas začátku a konce návštěvy stránky (s tím, že konec může být zavádějící) a navíc i některé informace zjištěné z těchto časů – den v týdnu, doba strávená na stránce, týden v roce, …

4.2 Analýza systému Moodle

Moodle je velký systém, který umožňuje uživatelům mnoho věcí změnit a nastavit. Z toho důvodu obsahuje velké množství tabulek. Moodle umožňuje přidávat různé moduly pro výuku a evidenci výsledků. Umožňuje například přidání chatů a diskusí, zobrazování studijních materiálů, ankety, úkoly a další. Jelikož je systém Moodle pod obecnou veřejnou licencí GNU, může si kdokoli upravit podle svých potřeb, nebo si vytvořit vlastní moduly.

Z pohledu data miningu nás bude zajímat jen velmi malá část tohoto systému a proto budu popisovat pouze ji.

4.2.1 Testy

V systému Moodle jsou testy vytvářeny z jednotlivých otázek, přičemž při vytváření testu je možné vybrat konkrétní otázky, nebo nastavit množinu otázek, ze kterých se má automaticky vybírat. Dále je možné nastavit časové období, po které je test přístupný, počet pokusů a další informace.

Je dostupné velké množství typů otázek. Při vytváření otázky je možné vybírat z jedenácti druhů, podle druhu odpovědi, nebo druhu otázky:

- Krátká tvořená odpověď – má krátkou odpověď v rozsahu jednotek slov, která je porovnávána s různými předpřipravenými modelovými odpověďmi.
- Numerická úloha – má jako výsledek číselnou hodnotu, která může být i s jednotkami. Vyhodnocení probíhá porovnáním zadané hodnoty s modelovými odpověďmi se zadanou tolerancí odchylky.
- Výběr z možných odpovědí – umožňuje vybrat jednu nebo více odpovědí ze seznamu možností.
- Pravda/Nepravda – je speciální případ výběru z možných odpovědí, kdy se vybírá jedna možnost (Pravda nebo Nepravda).

- Dlouhá tvořená odpověď – dovoluje odpovědět ve formě delšího textu (několik vět nebo odstavců) ale vyžaduje následně ruční vyhodnocení.
- Vypočítávaná úloha - je podobná numerické úloze, ale každý student dostane jiné zadání z určité množiny.
- Jednoduchá vypočítávaná úloha – je jednodušší variantou vypočítávané úlohy, je podobná numerické úloze, ale každý student dostane jiné zadání z určité množiny.
- Vypočítávaná úloha s více možnostmi – je podobná úloze s více možnostmi, ale nabízené odpovědi pro každého studenta se vygenerují automaticky podle vzorce
- Přiřazování – odpověď na každou dílčí úlohu je vybrána ze seznamu možností.
- Přiřazování pro náhodně vybrané – je stejné, jako přiřazování, ale úlohy jsou vybrány náhodně.
- Doplňovací úloha – umožňuje vytvořit složitější úlohu složenou z několika dílčích úkolů.

V oficiální dokumentaci systému Moodle jsou tři sekce, které se vztahují k testům. Programátoři je nazvali "Question types", "Questions" a "Quiz system". V části "Question types" jsou tabulky, které se vztahují k různým typům otázek. Obsahují informace o speciálních otázkách a jejich odpovědích a pro data-mining nejsou relevantní. Část "Questions" obsahuje hlavní informace o otázkách a odpovědích a v části "Quiz system" se nachází tabulky týkající se samotných testů a jejich výsledků.

Každý záznam v tabulce *question* obsahuje informace o jedné otázce. Jsou zde informace o tom, kým a kdy byla otázka vytvořena, respektive změněna. Dále je zde defaultní bodování za správné zodpovězení a penalizace za špatnou odpověď. Nedílnou součástí je typ otázky. Přestože systém umožňuje vytvořit více než deset různých druhů úloh, všechny jsou uloženy v této tabulce a musí tedy být rozlišeny (shortanswer, multichoice, essay, truefalse, …). Samozřejmě nesmí chybět název a text otázky.

Tabulka *question_answers* obsahuje všechny modelové odpovědi a odpovědi nabízené u úloh. Každý záznam obsahuje odkaz na otázku, se kterou je spojen a text odpovědi. Dále obsahuje informaci o tom, jakou částí známky za úlohu má být odpověď ohodnocena. To se hodí například u otázek s více správnými odpověďmi, nebo se díky tomu dají vytvořit modelové odpovědi, které budou částečně správné.

Záznamy v tabulce *quiz* obsahují informace o jednotlivých testech. V těchto záznamech jsou informace o testu, jako je název a nějaký popis, kurz, se kterým je daný test spojen, dále je zde určen čas začátku a konce, kdy je možné test vypracovávat, a časový limit testu. Dále je zde určeno omezení pro počet pokusů a jsou zde určeny otázky, které se v testu vyskytují. Nesmí chybět ani bodové ohodnocení celého testu a způsob vyhodnocení (jestli výslednou známkou z testu bude známka za poslední test, první test, nejlepší test, …).

Tabulka *quiz_question_instances* je vlastně vazební tabulkou mezi otázkami a testy. Kromě samotné vazby je zde i ohodnocení jednotlivých otázek v rámci testu.

4.2.2 Výsledky testů

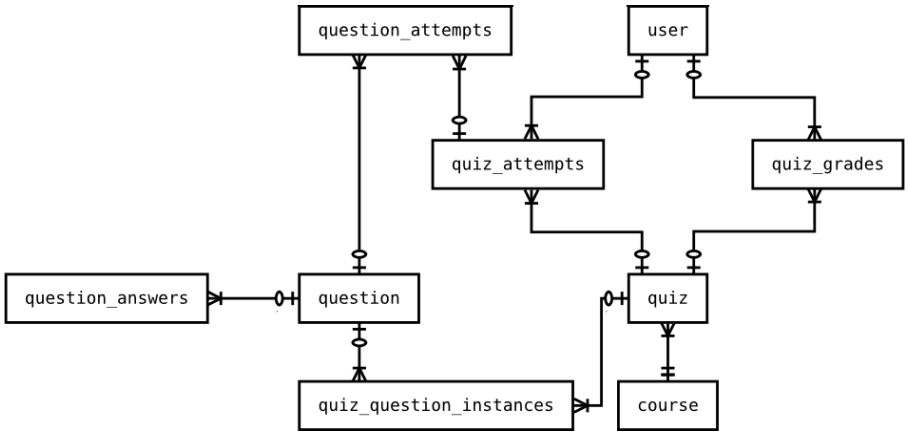
Po těchto tabulkách, které tvoří databázi otázek a testů, se můžeme dostat k tabulkám výsledků, které budou potřebné pro dolování dat. Jednou z nich je tabulka *quiz_attempts*. Záznamy v ní obsahují informace o uskutečněných testech. Obsahují tedy odkaz na test a na uživatele, který jej vyplňoval. Dále je zde čas začátku a konce testu, číslo pokusu a bodové hodnocení daného pokusu.

Tabulka *quiz_grades* obsahuje celkové hodnocení testu, ne jen pokusu. Záznamy obsahují odkazy na test a uživatele, společně s časem poslední změny záznamu a hodnocení testu.

Další tabulkou důležitou pro data mining je *question_attempts*. V této tabulce jsou záznamy o jednotlivých výskytech otázek v testech. Do této tabulky se ukládají správné odpovědi a odpovědi zadané uživateli. Také je zde uložena informace o tom, jaké bodové hodnocení měla daná otázka v testu. Defaultní nastavení se může měnit v rámci jednotlivých testů a stejně tak mohou být různě hodnoceny náhodně generované otázky, proto je potřeba tuto informaci zachovat u konkrétního řešení.

4.2.3 Záznamy přístupů

V systému Moodle se záznamy přístupů ukládají do tabulky *log*. V ní je uloženo jaký uživatel a kdy zobrazil stránku a z jaké IP adresy se připojil. Dále je zde uvedeno, jaký modul uživatel využíval (quiz, course, login, user, …), o jakou akci šlo (view, login, logout, attempt, editquestions, …) a jaká byla URL adresa dotazu.



Obrázek 5 *ER diagram části databáze systému Moodle, která je spjatá s testy a jejich výsledky*

5 Analýza požadavků

Tato kapitola se věnuje analýze požadavků na výslednou aplikaci. Tato analýza popisuje základní požadavky a představy o aplikaci. Je zde popsáno, k čemu bude výsledná aplikace sloužit, kdo s ní bude pracovat a jaká data bude používat.

5.1.1 K čemu má aplikace sloužit

Výsledná aplikace má umožnit analýzu dat v systému Barborka. Měl by umožnit rozvoj výuky i studia, což znamená, že s ním budou pracovat jak vyučující, tak studenti. Je tedy potřeba vytvořit samotný nástroj pro analýzu e-learningového systému a uživatelské rozhraní, které umožní studentům a vyučujícím vhodným způsobem využívat tuto aplikaci.

5.1.2 Kdo bude se systémem pracovat

Tato aplikace umožní přístup více uživatelům, ale ne všichni budou mít k dispozici stejné funkce. Je proto potřeba rozdělit je podle rolí.

- Student: Má možnost analyzovat své výsledky a porovnat je s výsledky ostatních studentů (tyto nejspíš ve formě průměru).
- Vyučující: Má možnost provádět analýzy předmětu, termínů testů, testů a otázek. U těchto analýz může upravovat nastavení parametrů a zvolit data miningovou metodu.

5.1.3 Vstupy

Aplikace přijme data z databáze e-learningového systému, zpracuje je a uloží je do vlastní databáze.

Pokud bude s aplikací pracovat student, vstupem jeho analýzu bude jeho identifikátor a identifikátor předmětu, který studuje.

V případě, že bude s aplikací pracovat vyučující, vstupem jeho analýzy bude metoda, kterou chce použít a její parametry, identifikátor kurzu, který chce analyzovat a podrobné informace o evidovaných informacích, které chce k analýze použít.

5.1.4 Výstupy

Aplikace zobrazí výsledek analýzy. Dále bude umožňovat zobrazení předešlých analýz. To vše v přehledné formě.

- Statistická analýza bude umožňovat zobrazení základních statistik v podobě grafů, stejně jako podrobnější informace v podobě tabulky
- Shluková analýza umožní zobrazení historie analýz a u každé analýzy bude možné zobrazit shluky a jejich obsah
- Analýza pomocí rozhodovacích stromů umožní zobrazení provedených analýz a názorné zobrazení vybraného rozhodovacího stromu

- Analýza pomocí asociačních pravidel umožní zobrazení provedených analýz a u vybrané analýzy bude možné zobrazit asociační pravidla ve formě tabulky

5.1.5 Funkce aplikace

- Aktualizace dat ze zdrojové databáze
- Statistická analýza pro studenty
- Statistická analýza
    - Zadání a editace parametrů
    - Zobrazení analýzy
- Shluková analýza
    - Zadání a editace parametrů
    - Zobrazení shlukování
- Analýza pomocí rozhodovacích stromů
    - Zadání a editace parametrů
    - Zobrazení a procházení rozhodovacího stromu
- Analýza pomocí asociačních pravidel
    - Zadání a editace parametrů
    - Zobrazení asociačních pravidel

5.1.6 Nefunkční požadavky

Aplikace bude realizována jako webová aplikace, respektive rozšíření systému eLogika. Kvůli jednoduššímu napojení na systém eLogika bude aplikace využívat databázi MS SQL a bude psaná v jazyce C#.

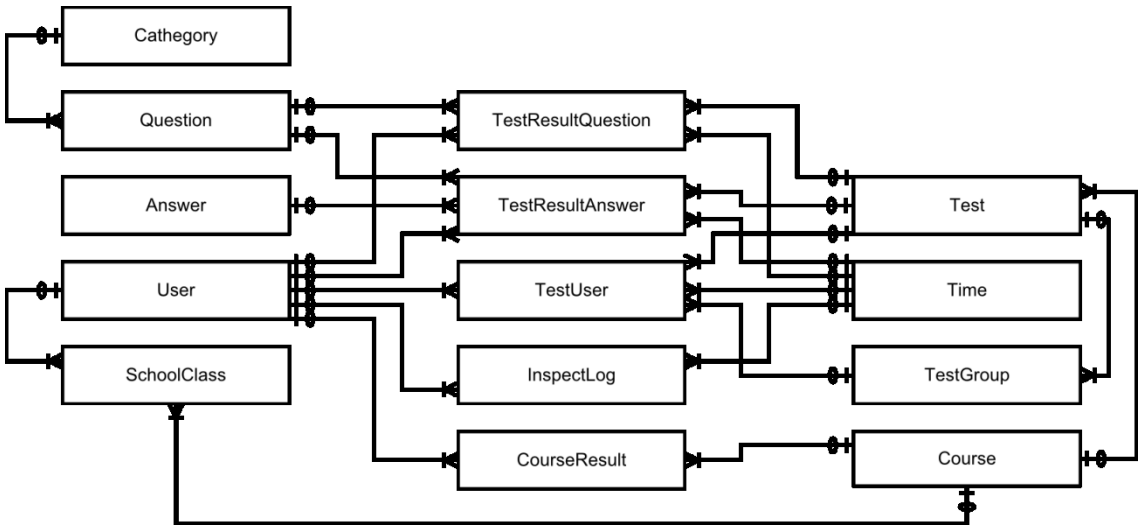
6 Návrh vyvíjené aplikace

V této kapitole se nachází návrh data miningové aplikace. Je zde návrh databáze, tedy datového skladu a také návrh samotné aplikace, tedy případy užití a s nimi spojené diagramy.

6.1 Návrh datového skladu

Návrh datového skladu je důležitý, protože předzpracování dat a jejich uložení v příhodné formě ovlivňuje rychlost a možnosti analýzy. V této kapitole tedy bude návrh struktury uložení dat. Teorii datových skladů je více popsaná v kapitole 2.1 a literatuře (3). Tento datový sklad může s drobnými úpravami sloužit pro analýzu systémů Barborka, Moodle i eLogika (viz (23)).

Při tvorbě datového skladu jsme vycházeli z databáze systému Barborka, proto se obsah a význam tabulek moc neliší. Přesto ale bylo nutné přidat tabulky, které v původní databázi nejsou a v jiných tabulkách přibyly atributy, které bylo možné dopočítat při importu a které jsou následně použity při analýzách. Diagram datového skladu je na obrázku Obrázek Obrázek 6.



Obrázek 6 *ER diagram datového skladu*

Jelikož je tato práce zaměřená i na systém Moodle, obsahují některé tabulky atributy, které se v systému Barborka neevidují, ale které bude možné použít, pokud by se měla výsledná aplikace použít pro analýzu systému Moodle. Datový slovník se všemi parametry tabulek je součástí přílohy.

Tabulka *Test* je dimenzionální tabulka, která obsahuje popis testu. Jsou zde tedy údaje jako maximum bodů, počet otázek a odhadovaný čas, nutný pro dokončení testu.

Tabulky *Question* a *Answer* jsou také dimenzionální. Obsahují informace o otázkách, resp. odpovědích. V tabulce *Question* je text otázky, identifikátor autora, vazba na kategorii otázky, vazba na kurz a identifikátor otázky z původní databáze. V tabulce *Answer* je vazba na

otázku, se kterou se pojí daná odpověď, text odpovědi, autor odpovědi a informace, jestli je odpověď správná nebo špatná.

Tabulka *Cathegory* obsahuje název kapitoly, do které otázka patří a umožňuje dělit otázky do skupin, případně analyzovat tyto skupiny otázek. Je to taky to jediné, co je možné zobrazit studentům při jejich zpětné vazbě.

Tabulka *TestGroup* je další dimenze, která reprezentuje testové skupiny, nebo termíny testů. Obsahuje tedy čas a místo konání testu. Dále ale obsahuje i pořadí termínu testu. Toto číslo udává, kolikrát se daný test konal. To by samozřejmě bylo možné zjistit i bez tohoto atributu, ale takto to bude jednodušší.

Tabulky *User* a *Course* jsou tabulky, které neposkytují moc informací. Tabulka *User* obsahuje identifikátor uživatele (ať už studenta, nebo vyučujícího) a jeho jméno a tabulka *Course* obsahuje název kurzu. Vazba kurzu s akademickým rokem je řešena pouze přes výsledky testů, protože pokud by existoval kurz bez výsledků, není možné jej analyzovat.

Tabulka *SchoolClass* zajišťuje vazbu mezi studentem a předmětem. Systém Barborka bohužel neeviduje studijní skupiny, takže není možné zjistit, kdo byl cvičící daného studenta. Oproti tomu v systému Moodle to již možné je, proto jsou v této tabulce atributy, které umožní evidovat takovouto informaci. Je tu tedy vazba na kurz, studenta, cvičícího, akademický rok a typ třídy pro případné rozlišení cvičení a přednášky.

Poslední dimenzionální tabulkou je tabulka *Time*. Jelikož v databázi Barborka jsou časové údaje reprezentovány pomocí řetězce, byl v datovém skladu vytvořen speciální objekt, který zjednoduší práci s časovými údaji. Oproti základním atributům (rok, měsíc, den, hodina, minuta, vteřina) jsou zde i atributy pro akademický rok a den v týdnu.

Mezi tabulky faktů patří *TestResultQuestion*. Záznamy v této tabulce reprezentují konkrétní zodpovězené otázky z testů. Je zde evidováno datum a čas řešení testu (ve formě reference na záznam v tabulce *Time*), maximum bodů za otázku, získaný počet bodů, kvadrát získaného počtu bodů, čas, který student strávil řešením otázky a jeho kvadrát. Zbývající atributy jsou reference na dimenzionální tabulky *Test* a *Question*.

Záznam v tabulce *TestResultAnswer* reprezentuje konkrétní odpověď uživatele. Eviduje se zde, jestli byla odpověď zodpovězena správně a reference na tabulky *Time*, *Test*, *Question* a *Answer*.

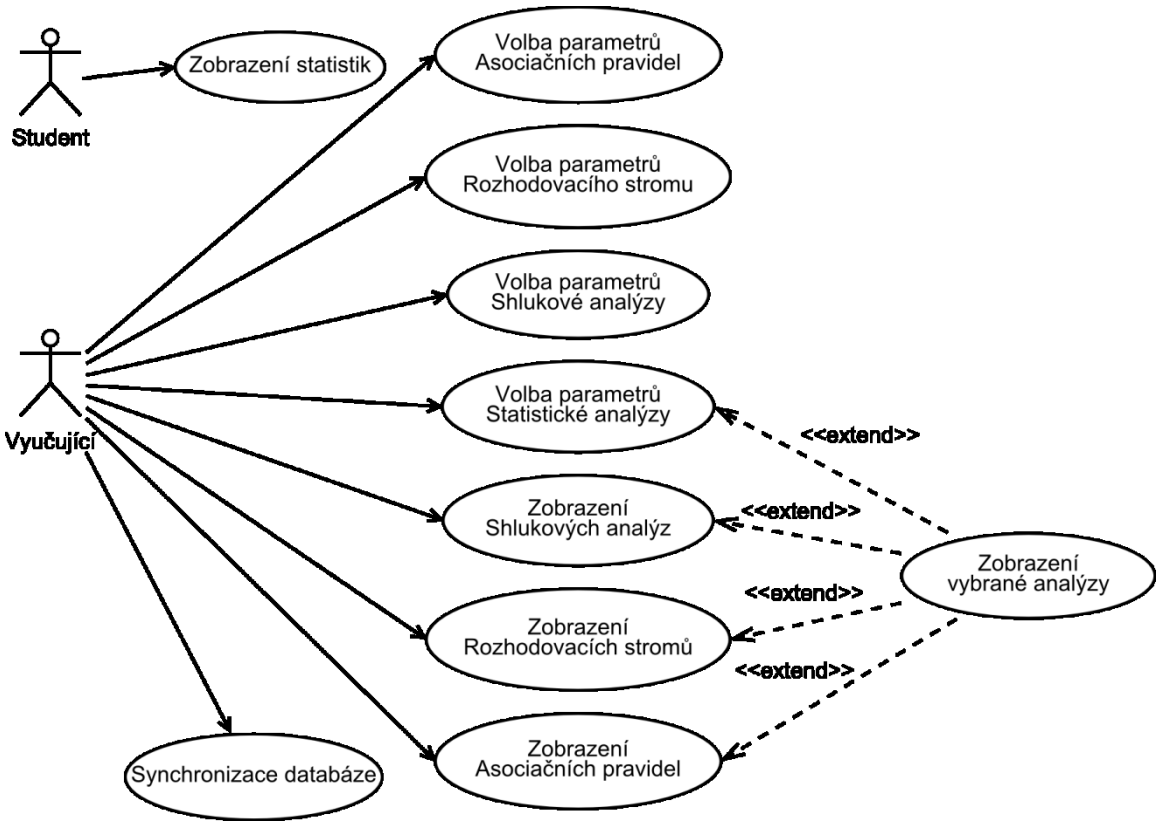
Tabulka *TestUser* obsahuje informace o konkrétních testech, které uživatelé řešili. Je zde doba, po kterou uživatel řešil test a její kvadrát, počet bodů, které získal a jejich kvadrát a číslo pokusu o zvládnutí testu. Nejedná se zde o číslo termínu testu, ale o počet termínů testu, kterých se uživatel již zúčastnil. Kromě toho jsou zde reference na tabulky *Test*, *User*, *Time* a *TestGroup*.

Tabulka *CourseResult* obsahuje souhrnné údaje za každé absolvování kuru studentem. Je zde reference na tabulky *User* a *Course*, označení akademického roku a počet bodů z evidovaných aktivit.

Poslední tabulkou faktů je *InspectLog*. Tato tabulka obsahuje záznamy přístupů do systému, tedy IP adresu uživatele, část systému, kterou navštívil, URL adresa, kurz, do kterého se přihlásil a reference na tabulky *User* a *Time*.

6.2 Případy užití

Případ užití je sekvence kroků, které definují interakci uživatele (aktéra) se systémem. V této podkapitole je uvedeno několik základních případů užití a diagram případu užití, který graficky znázorňuje funkce systému a práva uživatelů (nebo uživatelských rolí) používat tyto funkce.



Obrázek 7 *Use Case diagram*

6.2.1 Volba parametrů asociačních pravidel

Případ užití umožňuje vybrat atributy, které se mají použít pro generování asociačních pravidel. Umožňuje nastavit další parametry pro vytvoření těchto pravidel (spolehlivost a podporu).

Aktéři: vyučující, systém

Základní tok:

1. Systém zobrazí formulář a seznam předmětů, které jsou v databázi
2. Vyučující vybere předmět, kterých chce analyzovat

3. Systém zobrazí pro vybraný předmět akademické roky, ve kterých jsou evidovány záznamy v databázi
4. Vyučující zvolí akademické roky, a jaký objekt chce analyzovat (např.: Kategorie otázek, Otázky, Testy, Termíny, …)
5. Systém zobrazí atributy dostupné pro vybraný objekt
6. Vyučující zvolí atributy objektu, které chce použít pro analýzu a u každého zvolí, jestli se jedná o antecedenta nebo o konsekvent.
7. Vyučující zadá minimální podporu, minimální spolehlivost a zvolí, jestli se záznamy mají předzpracovat pomocí algoritmu quick ROCK a odešle formulář.
8. Systém zpracuje analýzu, výsledek uloží do databáze a zobrazí jej uživateli.

Podmínky pro dokončení:

Analýza je uložená v databázi

6.2.2 Volba parametrů rozhodovacího stromu

Případ užití umožňuje vybrat atributy, které se mají použít pro generování rozhodovacího stromu.

Aktéři: vyučující, systém

Základní tok:

1. Systém zobrazí formulář a seznam předmětů, které jsou v databázi
2. Vyučující vybere předmět, kterých chce analyzovat
3. Systém zobrazí pro vybraný předmět akademické roky, ve kterých jsou evidovány záznamy v databázi
4. Vyučující zvolí akademické roky, a jaký objekt chce analyzovat (např.: Kategorie otázek, Otázky, Testy, Termíny, …)
5. Systém zobrazí atributy dostupné pro vybraný objekt
6. Vyučující zvolí atributy objektu, které chce použít pro analýzu.
7. Vyučující označí právě jeden ze zvolených atributů jako predikovaný atribut.
8. Vyučující zvolí, jestli se záznamy mají předzpracovat pomocí algoritmu quick ROCK a odešle formulář.
9. Systém zpracuje analýzu, výsledek uloží do databáze a zobrazí jej uživateli.

Podmínky pro dokončení:

Analýza je uložená v databázi

6.2.3 Volba parametrů shlukové analýzy

Případ užití umožňuje vybrat atributy, které se mají použít pro generování rozhodovacího stromu.

Aktéři: vyučující, systém

Základní tok:

1. Systém zobrazí formulář a seznam předmětů, které jsou v databázi
2. Vyučující vybere předmět, kterých chce analyzovat
3. Systém zobrazí pro vybraný předmět akademické roky, ve kterých jsou evidovány záznamy v databázi
4. Vyučující zvolí akademické roky, a jaký objekt chce analyzovat (např.: Kategorie otázek, Otázky, Testy, Termíny, …)
5. Systém zobrazí atributy dostupné pro vybraný objekt
6. Vyučující zvolí atributy objektu, které chce použít pro analýzu.
7. Vyučující vybere shlukovací metodu a její parametry a odešle formulář.
9. Systém zpracuje analýzu, výsledek uloží do databáze a zobrazí jej uživateli.

Alternativní tok 1:

7.1 Vyučující vybral algoritmus k-means nebo hierarchické shlukování.

7.2 Vyučující vybere metodu měření vzdálenosti, počet shluků, které mají vzniknout, a zvolí, jestli má algoritmus hledat lepší analýzu pro jiný počet shluků.

Alternativní tok 2:

7.1 Vyučující vybral algoritmus quick ROCK.

7.2 Vyučující určí threshold pro určení hranice, kdy jsou si dva objekty ještě podobné.

7.3 Vyučující zvolí, jestli se mají numerická data kategorizovat automaticky, nebo pomocí metody k-means

Podmínky pro dokončení:

Analýza je uložená v databázi

7 Implementace

Tato kapitola je věnována samotné implementaci navržené aplikace. Jsou zde rozebrány hlavní části aplikace a její struktura. Pro vytvoření aplikace byla použita technologie ASP.NET 4.0 s využitím softwarové architektury MVC.

První část této kapitoly je věnována struktuře aplikace, jednotlivým vrstvám a pomocným objektům, které reprezentují jednotlivé analýzy. Další část je pak věnována složitějším objektům, které mohou vstupovat do analýz.

7.1 Struktura aplikace

Tato aplikace obsahuje 3 vrstvy, které mezi sebou komunikují. Datová vrstva (DAL) umožňuje práci s databází a při změně databáze stačí tedy udělat změny pouze v této vrstvě. Business vrstva (BAL) zpracovává data z datové vrstvy, dále je zpracovává (zde je využito data miningových metod) a výsledek předá prezentační vrstvě.

Celá aplikace je rozdělena do několika samostatných částí:

- DataMiningDAL: Část pro práci s daty. Tato část komunikuje přímo s databází
- DataMiningBAL: Tato část přijímá dotazy z webových formulářů a posílá dotazy na DataMiningDAL, ze které získá data a zpracuje je. Ke zpracování využívá knihovnu DataMiningLibrary.
- DataMiningLibrary: Jedná se o knihovnu, ve které jsou naimplementovány vybrané data miningové metody.
- DataMiningWeb: Toto je zobrazovací část. Obsahuje webové formuláře, které jsou zobrazovány uživatelům.

V aplikaci jsou kromě databázových objektů využívány i objekty, které reprezentují jednotlivé druhy data miningové analýzy a jejich části. Obrázek 8 obsahuje třídní diagram těchto objektů.

Shluková analýza je reprezentována objektem DMCluster. V tomto objektu je identifikován shlukovaný objekt a všechny parametry shlukování, které byly použity. Dále obsahuje pole objektů DMClusterNode, tedy pole jednotlivých shluků analýzy. Každý z těchto shluků obsahuje pole shlukovaných vektorů.

Rozhodovací strom je reprezentován objektem DMDecisionTree. V tomto objektu jsou všechny atributy rozhodovacího stromu a kořenový uzel stromu reprezentovaný objektem DMDesicionTreeNode. Každý uzel stromu obsahuje seznam atributů, které v něm jsou použity, atribut, který byl použit k jeho vytvoření a pole potomků.

Asociační pravidla jsou reprezentována objektem DMAssociations. V tomto objektu jsou všechny vlastnosti analýzy (hraniční podpora, hraniční spolehlivost). Dále je zde pole samotných pravidel (DMAssociationRule). Každé pravidlo obsahuje množinu příčin, množinu následků, podporu a spolehlivost.

Obrázek 8 *Třídní diagram pomocných objektů*

7.2 Analyzované objekty

Při použití statistických metod se analyzují objekty, které jsou v databázi, protože vybrané statistické metody pracují vždy pouze s jedním atributem. Například při výpočtu průměrného počtu bodů z testu se pracuje pouze s počtem bodů.

Pro účely data miningu je potřeba vytvořit složitější objekty, které umožní rozsáhlejší analýzy. Přestože u výsledku testu je informace o studentovi a jeho výsledku, je potřeba umožnit uživateli aplikace zvolit další relevantní atributy. Jedná se o atributy z tabulek dimenzí (viz kapitola 2.1 Datový sklad).

Například k výsledku testu je možné dohledat informace o termínu (čas, kapacita, pořadí, …), o testu (maximum bodů, autor testu, …), o předmětu, atd.

7.2.1 Otázky

Základním objektem pro analýzu je otázka. U otázky je možné zjistit kategorii, do které patří a jejího autora. Dále můžeme u každé otázky dopočítat průměrnou úspěšnost a počet použití. Průměrná úspěšnost je normovaná hodnota, kdy 1 značí, že otázka byla zodpovězena na 100% správně a 0 značí, že otázka byla zodpovězena naprosto nesprávně.

Jeden objekt reprezentuje jeden text zadání otázky.

7.2.2 Odpovědi

Dalším základním objektem pro analýzu je odpověď. U odpovědi známe otázku, ke které odpověď patří, autora odpovědi a správnost odpovědi. Dále můžeme dopočítat průměrnou úspěšnost otázky, počet použití otázky, počet použití odpovědi a úspěšnost odpovědi.

Průměrná úspěšnost je normovaná hodnota, kdy 1 značí, že otázka byla zodpovězena na 100% správně a 0 značí, že otázka byla zodpovězena naprosto nesprávně. Úspěšnost odpovědi udává poměr mezi počtem správných zodpovězení odpovědi (odpověď je správná a byla studentem zvolena, nebo naopak je odpověď nesprávná a zvolena nebyla) a počtem všech použití odpovědi.

Jeden objekt reprezentuje jeden text odpovědi.

7.2.3 Termíny

Dalším objektem je termín konání testu. U každého termínu můžeme zjistit počet studentů, kteří se termínu zúčastnili, pořadí termínu, datum a čas konání termínu, informace o testu (autor, identifikátor testu, počet bodů za test). Dále můžeme zjistit průměrnou úspěšnost termínu a průměrnou úspěšnost testu. Průměrné úspěšnosti udávají průměrný poměr mezi získanými body a maximem bodů za test.

Jeden objekt reprezentuje jeden termín

7.2.4 Varianty testů

Dalším objektem je test, respektive varianta testu. Varianta testu je konkrétní kombinace otázek a odpovědí. U takového varianty je dostupný identifikátor testu (což je v podstatě pouze šablona), počet otázek v testu a maximální počet bodů. Dále můžeme dopočítat počet použití testu a varianty testu a průměrnou úspěšnost testu a varianty. Průměrné úspěšnosti udávají průměrný poměr mezi získanými body a maximem bodů za test.

Jeden objekt reprezentuje jednu variantu testu.

7.2.5 Výsledky otázek

Výsledky otázek jsou dalším objektem, který je možné analyzovat. U každého použití otázky je dostupná informace o otázce (identifikátor otázky a autor otázky), dále identifikátor testu, ve kterém byla použita a typ testu (cvičný nebo hlavní). Dále máme informaci o výsledku použití otázky, tedy kolik bodů student získal.

Jeden objekt reprezentuje jedno použití otázky, tedy odpověď jednoho studenta na otázku.

7.2.6 Výsledky odpovědí

Dále máme k dispozici výsledky odpovědí. U každého použití odpovědi je dostupný identifikátor otázky, identifikátor odpovědi a identifikátor testu, ve kterém byla otázka a

odpověď použita a typ testu (cvičný nebo hlavní). Dále máme informaci o správnosti odpovědi (správná nebo nesprávná) a o jejím zodpovězení (zodpovězena správně nebo špatně).

Jeden objekt reprezentuje jedno použití odpovědi, tedy zobrazení odpovědi jednomu studentovi.

7.2.7 Výsledky testů

Dalším objektem je výsledek testu. U výsledku testu máme k dispozici identifikátor testu a studenta, který ho vypracovával, dále je zde číselné označení pokusu o absolvování testu (pořadí studentova pokusu) a typ testu (cvičný nebo hlavní). Jelikož je test vypracován vždy v nějakém termínu, je zde i informace o termínu, tedy počet studentů na termínu, čas konání termínu a číslo termínu. Poslední informací je získaný počet bodů za test.

Jeden objekt reprezentuje jedno vypracování testu studentem.

7.2.8 Výsledky podle kategorií

Posledním objektem je takzvaný výsledek kategorií. Jedná se průměrný výsledek studenta z otázek náležících do vybraných kategorií. U tohoto objektu se pracuje s identifikátorem studenta a s jeho průměrnými výsledky v jednotlivých kategoriích. Průměrný výsledek v kategorii je průměrný poměr mezi získaným počtem bodů a maximálním počtem bodů ze všech otázek, které student vypracoval a které zároveň patří do dané kategorie.

Příklad: Pokud jsou ve zkoumaném předmětu dvě kategorie otázek (K1, K2) a student absolvoval test, ve kterém byly z každé kategorie dvě otázky (K1-Otázka1, K1-Otázka20, K2-Otázka12, K2-Otázka22), pak jeho průměrný výsledek v kategorii K1 je průměrný výsledek z otázek K1-Otázka1 a K1-Otázka20 a obdobně, průměrný výsledek v kategorii K2 je průměrný výsledek z otázek K2-Otázka12 a K2-Otázka22.

Jeden objekt reprezentuje výsledky jednoho studenta.

8 Použití data miningu a interpretace výsledků

Po vytvoření funkční aplikace pro analýzu dat ze systému Barborka bude aplikace vyzkoušena a výsledky interpretovány. Tato část se nesoustředí na analýzu jednoho předmětu, ale na ukázky práce s data miningovou aplikací a její použití při analýze předmětů evidovaných v systému Barborka.

Použití data miningové aplikace se liší podle toho, kdo a co chce pomocí něj analyzovat. V této kapitole bude uvedeno použití aplikace studenty, kteří mají umožněno omezené použití aplikace. Dále zde budou uvedeny možnosti použití vyučujícími.

8.1 Zpětná vazba studentům

Jak bylo uvedeno dříve, zpětná vazba studentům je omezena a umožňuje pouze analýzu výsledků studia. Například není žádoucí umožnit studentovi vidět texty otázek a odpovědí. Proto je možné zobrazit pouze souhrnné informace za celou kapitolu, nebo za celý předmět.

Studentská část tedy umožňuje zobrazit průměrné výsledky přihlášeného studenta s cílem upozornit jej na kapitoly, které mu dělají potíže. Studentovi bude zobrazena tabulka se správností jeho odpovědí v jednotlivých kategoriích, průměrnou správností odpovědí studentů aktuálního ročníku a průměrnou správností odpovědí studentů za všechny vybrané roky. Tabulka má čtyři sloupce:

- Kategorie: identifikuje kategorii otázek.
- Průměr studenta v kategorii: je průměrná správnost studentových odpovědí na otázky v kategorii. Je to průměrná správnost zodpovězení otázek, které student vypracoval a které zároveň patří do dané kategorie.
- Průměr ročníku v kategorii: je průměrná správnost odpovědí na otázky v kategorii za všechny studenty v ročníku.
- Celkový průměr v kategorii: je průměrná správnost odpovědí na otázky v kategorii za všechny studenty, kteří kdy řešili nějakou otázku z dané kategorie.

Příklad

Výsledky analýzy vybraného studenta z předmětu "ANGL_EKF_1" jsou v Tabulka 511. Student může porovnat své výsledky s výsledky celého ročníku a s výsledky za všechny roky evidování kurzu. Student se po přečtení tabulky dozví, že jedinou kategorií otázek zvládá excelentně, v několika kategoriích je průměrný, ale v dalších má velké problémy. Například všechny otázky z kapitoly "Chapter 01" zodpověděl na 100%. Oproti tomu průměrná správnost jeho odpovědí na otázky z "Chapter 06" je 7,5%, takže častokrát odpovídá velmi špatně a měl by se na tuto kapitolu zaměřit nejvíce.

Kromě vlastních výsledků zde najde také průměrné výsledky aktuálního ročníku a průměrné výsledky všech evidovaných ročníků. Toto sice nejsou nezbytné informace

pro zlepšení jeho výuky, ale z pohledu motivace není na škodu, když se student dozví, zda mají ostatní studenti podobné problémy.

Kategorie	Průměr Studenta (%)	Průměr ročníku (%)	Celkový průměr (%)
Chapter 01	100,00	75,74	81,23
Chapter 02	50,00	65,56	70,28
Chapter 03	63,33	64,75	72,69
Chapter 04	26,00	45,13	47,34
Chapter 05	28,00	48,36	47,07
Chapter 06	7,50	46,30	46,45
Chapter 07	13,33	44,78	55,82
Chapter 08	56,67	59,14	64,40

Tabulka 5 *Výsledek analýzy dostupné studentům. Byly analyzovány výsledky náhodného studenta v předmětu ANGL_EKF_1*

Tento student má možnost zjistit, v jakých kategoriích má nejhorší výsledky ("Chapter 06" a "Chapter 07") a zaměřit se na ně. Tuto informaci může student využít v několika případech:

- Pokud student opakuje předmět
- Pokud má student možnost opakovat test, ve kterém je zkoušena látka z těchto kapitol
- Pokud jsou tyto kapitoly použity v několika testech v průběhu studia předmětu

8.2 Zpětná vazba vyučujícím

Oproti zpětné vazbě studentům, vyučující může využívat většího množství data miningových metod. Vyučující se může zaměřit na různé oblasti analýzy. V první řadě ho mohou zajímat výsledky eduminingu ve vztahu ke studentům. Dále může analyzovat kvalitu otázek a odpovědí v testech. V této kapitole je rozebráno několik otázek, které si vyučující může položit a které mu data miningová aplikace může pomoci zodpovědět.

Lze vyhledat skupiny studentů s podobnými problémy?

Takovouto otázku si vyučující může položit například ve chvíli, kdy chce vytvořit různé cvičné testy a chce je zaměřit na kapitoly, ve kterých byli studenti méně úspěšní. Průměrné výsledky kategorií dokáže zjistit pomocí statistických metod, ale díky shlukování je vyučující schopen jednoduše vybrat skupiny studentů, které mají problém s nějakou kombinací kapitol.

K tomuto účelu se hodí shluková analýza výsledků kategorií otázek. Vyučující zvolí roky a kategorie, které chce analyzovat a následně zvolí metodu shlukování.

- V případě metody k-means nebo hierarchické shlukování je třeba vyzkoušet několik analýz s různým počtem shluků v závislosti na počtu analyzovaných kategorií a na výsledcích studentů.
- U metody QuickROCK je potřeba jako parametr threshold zvolit hodnotu 1. Takto vzniknou shluky studentů, kteří mají po kategorizaci stejné výsledky.

	Shluk	Počet objektů	(| Chapter 01 | Chapter 02 | Chapter 03 | Chapter 04)
Výběr	Cluster 0	38	(1,00 <= x <= 1,00| 0,67 <= x <= 0,88| 0,83 <= x <= 1,00| 0,00 <= x <= 0,50)
Výběr	Cluster 1	36	(1,00 <= x <= 1,00| 0,67 <= x <= 0,88| 0,83 <= x <= 1,00| 0,55 <= x <= 0,80)
Výběr	Cluster 2	30	(1,00 <= x <= 1,00| 1,00 <= x <= 1,00| 0,83 <= x <= 1,00| 0,55 <= x <= 0,80)
Výběr	Cluster 3	11	(1,00 <= x <= 1,00| 0,67 <= x <= 0,88| 0,44 <= x <= 0,75| 0,00 <= x <= 0,50)
Výběr	Cluster 4	9	(1,00 <= x <= 1,00| 1,00 <= x <= 1,00| 0,83 <= x <= 1,00| 0,00 <= x <= 0,50)
Výběr	Cluster 5	8	(1,00 <= x <= 1,00| 0,00 <= x <= 0,63| 0,83 <= x <= 1,00| 0,55 <= x <= 0,80)
Výběr	Cluster 6	8	(1,00 <= x <= 1,00| 0,00 <= x <= 0,63| 0,83 <= x <= 1,00| 0,00 <= x <= 0,50)
Výběr	Cluster 7	6	(0,00 <= x <= 0,50| 0,00 <= x <= 0,63| 0,00 <= x <= 0,33| 0,00 <= x <= 0,50)
Výběr	Cluster 8	6	(1,00 <= x <= 1,00| 0,00 <= x <= 0,63| 0,44 <= x <= 0,75| 0,00 <= x <= 0,50)
Výběr	Cluster 9	6	(1,00 <= x <= 1,00| 1,00 <= x <= 1,00| 0,83 <= x <= 1,00| 0,90 <= x <= 1,00)
Výběr	Cluster 10	6	(1,00 <= x <= 1,00| 0,67 <= x <= 0,88| 0,83 <= x <= 1,00| 0,90 <= x <= 1,00)
Výběr	Cluster 11	4	(0,00 <= x <= 0,50| 0,67 <= x <= 0,88| 0,83 <= x <= 1,00| 0,55 <= x <= 0,80)

Obrázek 9 *Analýza výsledků kategorií metodou QuickROCK s threshold=1*

Na obrázku (Obrázek 9) je část výsledku shlukové analýzy metodou QuickROCK. Byly analyzovány výsledky ze čtyř kapitol v předmětu ANGL_EKF_1. Metoda QuickROCK vygenerovala 34 shluků. Vyučujícího může zajímat zejména několik shluků s velkým počtem objektů (v tomto případě studentů). Cluster 0 obsahuje 38 studentů, přičemž všichni odpovídali velmi správně na otázky z kategorií "Chapter 01" a "Chapter 03". Otázky v kategorii "Chapter 02" odpovídali správně na 67% až 88% a s kategorií "Chapter 04" mají velké problémy. Obdobně Cluster 2 obsahuje studenty, kteří druhou kategorii zvládají lépe, ale mají problémy se čtvrtou kategorií.

	Shluk	Počet objektů	(| Chapter 01 | Chapter 02 | Chapter 03 | Chapter 04)
Výběr	Cluster 0	73	(0,99| 0,96| 1,00| 0,78)
Výběr	Cluster 1	67	(0,93| 0,84| 0,98| 0,59)
Výběr	Cluster 2	38	(0,98| 0,71| 0,57| 0,42)
Výběr	Cluster 3	17	(0,42| 0,79| 0,98| 0,58)
Výběr	Cluster 4	5	(0,22| 0,17| 0,29| 0,06)

Objekty	Chapter 01	Chapter 02	Chapter 03	Chapter 04
Id studenta: 20	0,50	0,00	0,25	0,00
Id studenta: 39	0,00	0,50	0,50	0,00
Id studenta: 1244	0,11	0,11	0,11	0,12
Id studenta: 1298	0,00	0,00	0,25	0,00
Id studenta: 2403	0,50	0,25	0,33	0,20

Obrázek 10 *Analýza výsledků kategorií metodou k-means*

Při analýze stejných dat metodou k-means (Obrázek 10) s využitím možnosti určení nekvalitnějšího shlukování jsme získali 5 shluků. Na obrázku je printscreen výsledné shlukové analýzy a obsah posledního shluku. Výsledkem je, že studenti v Cluster 0 mají problémy se čtvrtou kapitolou, Cluster 1 má horší výsledky druhé a čtvrté kapitolou, Cluster 2 má dobré

výsledky pouze v první kapitole, Cluster 3 má naopak dobré výsledky pouze ve třetí kapitole a poslední skupina studentů má nejhorší výsledky ve všech kapitolách.

Metoda k-means umožňuje uvnitř shluků určitou variabilitu, takže studenti ve stejném shluku nemají úplně stejné výsledky, ale na rozdíl od metody QuickROCK je možné získat menší počet shluků. Obě tyto metody rozdělí studenty do skupin, ale každá vytvoří trochu jiné skupiny. V obou výsledných analýzách se ale vyskytují shluky studentů, které si odpovídají. Například v obou analýzách jsou identifikovány skupiny studentů, kteří mají potíže s kategoriemi "Chapter 02" a "Chapter 04".

Následně může vyučující reagovat na kombinace kategorií, které dělají problém dostatečně velké skupině studentů. Může se na dané kategorie více zaměřit při výuce, nebo může vytvořit cvičné testy na jejich procvičení.

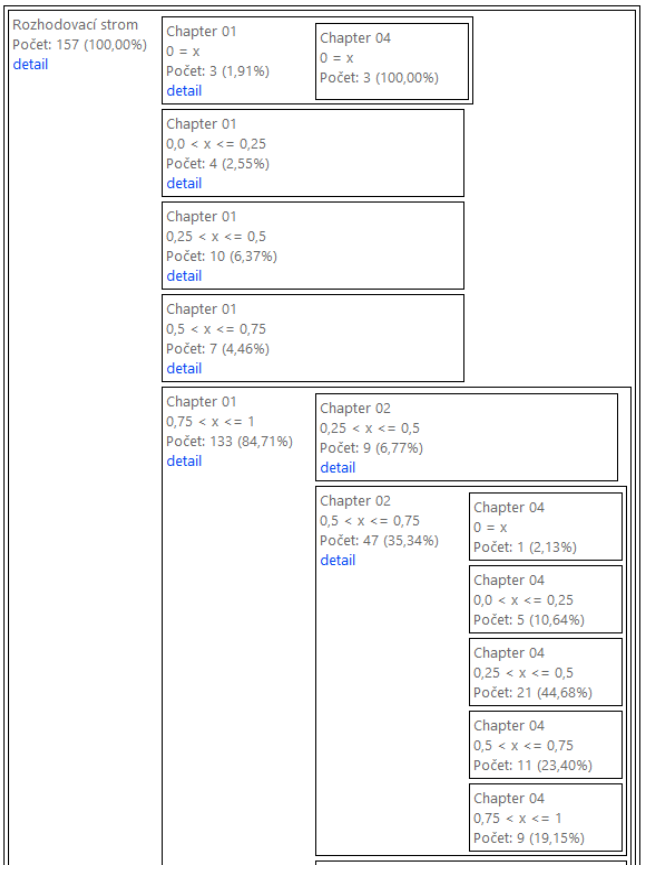
Jaká je intenzita vztahu mezi kategoriemi otázek?

Po zjištění, že existují skupiny kategorií, které jsou obtížné pro určité skupiny studentů, nás zajímá intenzita závislostí mezi kategoriemi otázek. K tomuto účelu se hodí asociační pravidla a rozhodovací stromy. Každá z těchto metod umožňuje jiný pohled na analýzu kategorií a je vhodné je kombinovat. Jako první můžeme použít metodu asociačních pravidel a následně nalezená pravidla ověřit metodou rozhodovacích stromů.

Podpora	Spolehlivost	Antecedent	Konsekvent
97 (47,32%)	57%	Chapter 01 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
93 (45,37%)	58%	Chapter 03 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
86 (41,95%)	61%	Chapter 01 : 0,75 < x <= 1 Chapter 03 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
72 (35,12%)	45%	Chapter 03 : 0,75 < x <= 1	Capter 07 : 0,75 < x <= 1
70 (34,15%)	64%	Chapter 02 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
68 (33,17%)	40%	Chapter 01 : 0,75 < x <= 1	Chapter 04 : 0,25 < x <= 0,5
68 (33,17%)	40%	Chapter 01 : 0,75 < x <= 1	Capter 07 : 0,75 < x <= 1
66 (32,20%)	67%	Chapter 02 : 0,75 < x <= 1 Chapter 03 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
65 (31,71%)	65%	Chapter 01 : 0,75 < x <= 1 Chapter 02 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
63 (30,73%)	45%	Chapter 01 : 0,75 < x <= 1 Chapter 03 : 0,75 < x <= 1	Capter 07 : 0,75 < x <= 1
62 (30,24%)	67%	Chapter 01 : 0,75 < x <= 1 Chapter 02 : 0,75 < x <= 1 Chapter 03 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
61 (29,76%)	36%	Chapter 01 : 0,75 < x <= 1	Chapter 05 : 0,75 < x <= 1
60 (29,27%)	38%	Chapter 03 : 0,75 < x <= 1	Chapter 05 : 0,75 < x <= 1
59 (28,78%)	35%	Chapter 01 : 0,75 < x <= 1	Capter 07 : 0,5 < x <= 0,75
54 (26,34%)	38%	Chapter 01 : 0,75 < x <= 1 Chapter 03 : 0,75 < x <= 1	Chapter 05 : 0,75 < x <= 1

Obrázek 11 *Analýza výsledků kategorií metodou asociačních pravidel*

Při tvorbě asociačních pravidel vyučující vybere roky a kategorie, které chce analyzovat. Následně zvolí hraniční podporu a hraniční spolehlivost. Obě tyto hodnoty se



Obrázek 12 *Analýza výsledků kategorií metodou rozhodovacích stromů*

zadávají v procentech. Volbou těchto atributů se zabývala kapitola 3.4.1. U každé zvolené kategorie otázek je potřeba určit, jestli se jedná o antecedent (příčinu) nebo o konsekvent (následek). Je možné zvolit obě možnosti zároveň.

Při použití metody rozhodovacích stromů vyučující zvolí roky a kategorie, které chce analyzovat. Z analyzovaných kategorií vybere vyučující jednu, jejíž výsledek se bude predikovat. Proto je vhodné předtím použít metodu asociačních pravidel, která naznačí možné vztahy.

Obrázek 11 ukazuje část asociačních pravidel pro analýzu výsledků kategorií (předmět ANGL_EKF_1). Byly použity všechny kategorie v předmětu, tedy kapitoly 01-08. Výsledky ukazují na to, že zvládnutí prvních kapitol by mohlo mít vliv na zvládnutí osmé kapitoly. Proto tuto domněnku potvrdíme ještě rozhodovacím stromem.

Kromě toho je jedním z výsledků pravidlo, že část studentů s dobrým výsledkem z "Chapter 01" má špatný výsledek z "Chapter 04". Toto pravidlo má poměrně velkou podporu, ale malou spolehlivost. Proto jsme použili metodu asociačních pravidel a rozhodovacích stromů i pro kapitoly 01-04. Výsledný rozhodovací strom je příliš velký, ale je možné zde zobrazit jeho část (Obrázek 12). Pro tuto analýzu byla použita data z předmětu ANGL_EKF_1. Byly použity kategorie 01-04 a "Chapter 04" byla zvolena pro predikci.

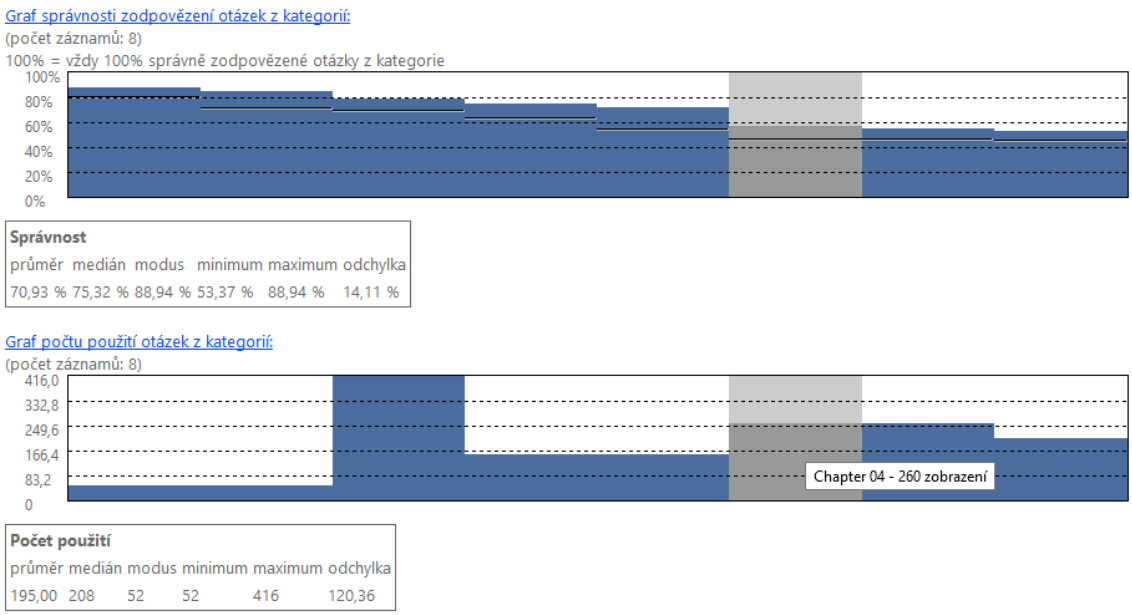
Z výsledného stromu je patrné, že "Chapter 03" nemá na "Chapter 04" vliv. Také se zde ukazuje, že obrovské procento studentů mělo výborné výsledky z "Chapter 01". Tímto jsou ovlivněny výsledky asociačních pravidel a proto se výborný výsledek z "Chapter 01" objevuje v mnoha pravidlech jako antecedent. Při dalším zkoumání výsledků se ukázalo, že to samé platí i pro "Chapter 03".

Jaká je kvalita otázek (kategorií, odpovědí) v systému?

Vyučující by měl začít analýzu předmětu tím, že zjistí kvalitativní rozložení kategorií, otázek a odpovědí v systému. K takovéto základní analýze stačí statistické metody, které umožní identifikovat velmi úspěšné nebo naopak velmi neúspěšné kategorie (nebo otázky nebo odpovědi).

Nalezení otázek, které studenti zodpovídají špatně, je důležité z několika důvodů. Za prvé je možné, že některé z těchto otázek obsahují chyby nebo jsou špatně formulovány. Takovéto otázky musejí být opraveny. Za druhé je možné, že látka, které se tyto otázky týkají, nebyla podána dostačujícím způsobem. V tom případě by bylo vhodné věnovat se v dalších letech této látce více, intenzivněji nebo z jiného úhlu pohledu.

Při statistické analýze kategorií je potřeba zvolit filtry, které umožní analyzovat výsledky použité v hlavních nebo cvičných testech. V případě systému Barborka nemají další filtry uplatnění, ale v jiných systémech, které evidují více dat, je možné filtrovat výsledky studentů vybraných cvičících, nebo například výsledky studentů v určité formě studia (prezenční/kombinovaní).



Obrázek 13 *Statistická analýza kategorií otázek*

Při statistické analýze otázek je kromě výše zmíněných filtrů možné analyzovat otázky použité ve vybraných testech. U statistické analýzy otázek a odpovědí může vyučující zvolit kategorie otázek, které chce analyzovat.

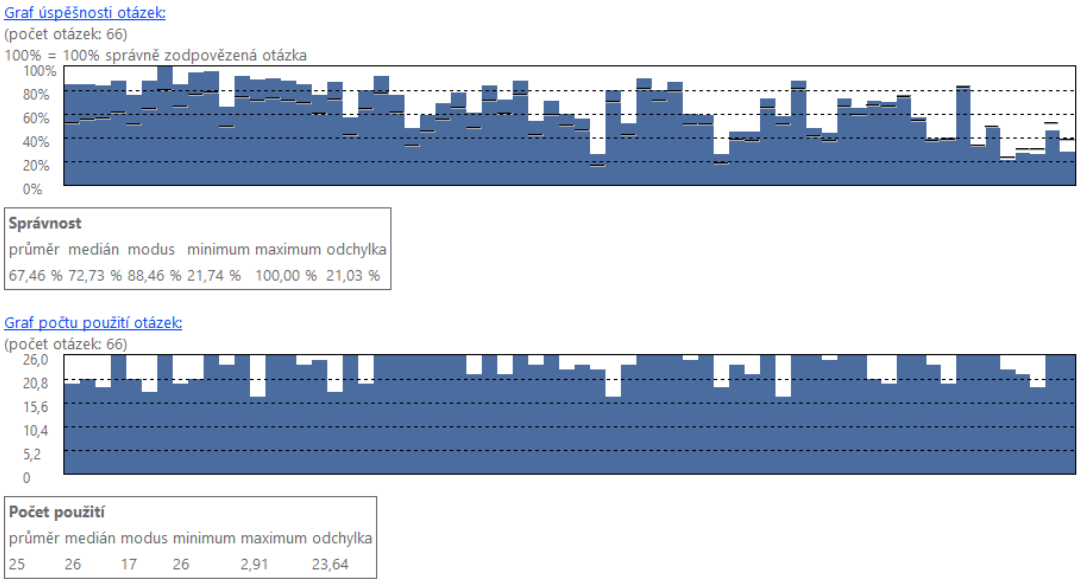
Dále je zde možnost zvolit hraniční správnost zodpovězení otázky (nebo odpovědi) a hraniční počet použití. Takto může vyučující získat pouze otázky, které byly použity dostatečně krát, protože analyzovat uspokojivě otázku, která byla použita pouze jednou, nelze. Hraniční správnost zodpovězení slouží k vyhledání příliš neúspěšných, nebo naopak úspěšných otázek.

Obrázek 13 ukazuje statistickou analýzu kategorií z předmětu ANGL_EKF_1. Statistickou analýzou získáme 2 grafy s tabulkami. V horním grafu je vyobrazena průměrná správnost zodpovězení otázek. Modrý sloupec udává správnost v analyzovaných letech a černá čára v každém sloupci označuje průměr za vybrané referenční roky. Díky tomu vyučující vidí, zda studenti se oproti předcházejícím rokům zlepšili. Ve spodním grafu je vyobrazen počet použití otázek z dané kategorie. U každého grafu je také tabulka se základními statistickými hodnotami pro všechny použité kategorie, jako je průměr, medián, modus, minimum, maximum a odchylka.

V tomto kurzu je 8 kategorií otázek, které jsou poměrně vyvážené. Není zde žádná kategorie, jejíž otázky by byly zodpovídány příliš špatně. Hranici si musí určit každý vyučující sám, ale dá se říct, že otázky, které studenti průměrně zodpovídají na méně, než 20% jsou buď špatně formulované, nebo velmi obtížné.

Statistická analýza jednotlivých otázek vychází relativně rovnoměrně (Obrázek 14). Grafy jsou setříděny podle rozdílu mezi správností zodpovídání otázek ve vybraném roce a v referenčních letech. Vlevo jsou tedy otázky, které studenti zodpovídali lépe, než v minulých letech a vpravo jsou naopak otázky, které zodpovídali méně správně. Na těchto grafech vidíme, že otázky jsou používány poměrně rovnoměrně a stejně rovnoměrná je úspěšnost jejich zodpovídání.

Pokud by se chtěl vyučující na něco zaměřit, mohl by zkontrolovat několik velmi úspěšných otázek (otázka se 100% průměrnou správností, která byla použita 26 krát je možná až příliš jednoduchá) nebo naopak neúspěšných otázek (je zde 6 otázek s průměrnou správností 20-25%).



Obrázek 14 *Statistická analýza otázek*

Závěr

Na začátku této práce je popsáno, co je to dobývání znalostí z databází a že data mining je pouze jeden z kroků při dobývání znalostí. V této souvislosti je zde popsáno využití data miningových metod v několika oblastech, zejména v marketingu a e-learningu.

Dále jsou zde popsány základní data miningové metody, které se dají použít pro edumining ze systémů Barborka a Moodle. Kromě základních statistických metod jsou zde popsány některé shlukovací algoritmy a algoritmy pro tvorbu rozhodovacích stromů a asociačních pravidel.

V rámci této práce měla být vyhodnocena možnost eduminingu ze systémů Barborka a Moodle. Proto jsou zde popsána data, která jsou těmito systémy evidována a která jsou vhodná pro edumining za účelem rozvoje výuky a studia. V přímé návaznosti na to byla navržena struktura datového skladu pro uložení potřebných dat z těchto systémů.

Během tvorby této diplomové práce byla také vyvinuta aplikace umožňující analýzu dat ze systému Barborka. Pomocí této aplikace byla analyzována data několika předmětů, které jsou evidované v systému Barborka. Tyto analýzy umožňují studentům identifikovat kategorie otázek, které jim dělají problémy a na které by se měli při studiu více zaměřit. Vyučujícím je tímto poskytnut nástroj, který mohou využít například ke kontrole otázek a identifikaci látky, která dělá studentům problémy. Dále mohou vyhledat skupiny studentů s podobnými problémy a zaměřit se na tyto problémy při tvorbě cvičných testů a cvičných příkladů.

Do budoucna by bylo možné vytvořit nástroj pro automatické generování cvičných testů přímo podle potřeb studenta. Jednalo by se o navázání na zpětnou vazbu studentům. Dále je zde prostor pro grafické znázornění shlukové analýzy. Jelikož je ve vzniklé aplikaci možné shlukovat n-rozměrná data, která mohou mít numerické i kategoriální atributy, nejedná se o triviální problém.

Použitá literatura

1. **Shaw, Michael J.** Knowledge management and data mining for marketing. *Decision Support Systems.* 2001, Sv. 1, 31, stránky 127-137. DOI 10.1016/S0167-9236(00)00123-8.

2. **Jain, A. K., Murty, M. N. a Flynn, P. J.** Data clustering: a review. *ACM Computing Surveys.* 1999, Sv. 3, 31, stránky 264-323. DOI 10.1145/331499.331504.

3. **Berry, Michael J. A. a Linoff, Gordon S.** *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management.* místo neznámé : Wiley, 2004. ISBN 0471470643.

4. **Romero, Cristobal a Ventura, Sebastián.** Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications.* 2007, Sv. 1, 33, stránky 135-146. DOI 10.1016/j.eswa.2006.04.005.

5. **Merceron, Agathe a Yacef, Kalina.** Educational data mining: A case study. *Artificial intelligence in education: Supporting learning through intelligent and sociall informed technology.* místo neznámé : IOS Press, 2005, stránky 467-474. DOI 10.1504/IJKESDP.2009.022718.

6. **Romero, Cristóbal a Ventura, Sebastián.** Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews.* 2010, Sv. 6, 40, stránky 601-618. DOI 10.1109/TSMCC.2010.2053532.

7. **Hoffman, P., a další, a další.** DNA visual and analytic data mining. *Proceedings. Visualization '97.* 1997.

8. **Antonie, M., Zaiane, O. a Coman, A.** Application of data mining techniques for medical image classification. *MDM/KDD.* 2001, stránky 94--101.

9. **Wang, Zhen a Yang, Meng.** A fast clustering algorithm in image segmentation. 2010, Sv. 6. DOI 10.1109/ICCET.2010.5486041.

10. **Berka, Petr.** *Dobývání znalostí z databází.* místo neznámé : Academia, 2003. str. 370. ISBN 8020010629.

11. **Han, Jiawei a Kamber, Micheline.** *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems).* místo neznámé : Morgan Kaufmann, 2006. ISBN 1558609016.

12. **Baker, R. S. J. d.** Data Mining. *International Encyclopedia of Education.* místo neznámé : Elsevier, 2010, Sv. 7, stránky 112-118.

13. **Otipka, P. a Šmajstrla, V.** *Pravděpodobnost a statistika*. [Online] 2008. [Citace: 1. 4 2014.] http://homen.vsb.cz/ oti73/cdpast1/index.htm. 80-248-1194-4.

14. **Litschmannová, Martina.** *Úvod do statistiky.* [Online] 2011. [Citace: 1. 4 2014.] http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/uvod_do_statistiky.pdf.

15. **Wu, Xindong, a další, a další.** Top 10 algorithms in data mining. *Knowledge and Information Systems.* 2007, Sv. 14, stránky 1-37. DOI 10.1007/s10115-007-0114-2.

16. **Deza, Michel Marie a Deza, Elena.** *Dictionary of Distances.* místo neznámé : Elsevier Science, 2006. ISBN 0444520872.

17. **Guha, Sudipto, Rastogi, Rajeev a Shim, Kyuseo.** Rock: a robust clustering algorithm for categorical attributes. *Information Systems.* 2000, Sv. 5, 25, stránky 345-366. DOI 10.1016/S0306-4379(00)00022-3.

18. **Dutta, M., Mahanta, A. Katoni a Pujari, Arun K.** QROCK: A quick version of the ROCK algorithm for clustering of categorical data. *Pattern Recognition Letters.* 2005, Sv. 15, 12, stránky 2364-2373. DOI 10.1016/j.patrec.2005.04.008.

19. **Jin, Chen, De-lin, Luo a Fen-xiang, Mu.** An improved ID3 decision tree algorithm. *2009 4th International Conference on Computer Science & Education.* 2009. DOI 10.1109/ICCSE.2009.5228509.

20. **Chang, Rui a Liu, Zhiyi.** An improved apriori algorithm. *Proceedings of 2011 International Conference on Electronics and Optoelectronics.* 2011, Sv. 1, stránky V1-476-V1-478. DOI 10.1109/ICEOE.2011.6013148.

21. **Menšík, Marek a Gerlich, Jakub.** *Data-mining and the quality of distance-education improvement.* Sofia : STEF92 Technology Ltd., 2013. SGEM 2013 : 13th international multidisciplinary scientific geoconference : GeoConference on Ecology, Economics, Education and legislation : 16-22, June, 2013, Albena, Bulgaria : conference proceedings. [Book 5]. Volume I. DOI 10.5593/SGEM2013/BB2.V1/S07.010. ISBN 978-619-7105-04-9.

Seznam příloh

Součástí DP je CD, na kterém jsou zdrojové kódy aplikace a programátorská dokumentace.

Příloha A: *Datový slovník*

Answer	typ	velikost	PK	null	def.	popis
Id	nvarchar	20	1	0		ID odpovědi (z původní databáze)
IdQuestion	nvarchar	20	0	1		ID otázky (Question.Id)
Text	nvarchar	max	0	1		Text odpovědi
AuthorId	int		0	0	-1	ID autora odpovědi (User.Id)
CourseId	nvarchar	20	0	1		ID kurzu (Course.Id)
Correct	bit	1	0	0	0	Správnost odpovědi

Tabulka A.Answer: Atributy tabulky Answer

Cathegory	typ	velikost	PK	null	def.	popis
Id	nvarchar	20	1	0		ID kategorie (z původní databáze)
Type	nvarchar	20	0	1		Název kategorie

Tabulka A.Cathegory: Atributy tabulky Cathegory

Course	typ	velikost	PK	null	def.	popis
Id	nvarchar	20	1	0		ID kurzu (z původní databáze)
Name	nvarchar	max	0	1		Název kurzu
Created_Ticks	bigint		0	1		Datum a čas vytvoření (Time.Ticks)

Tabulka A.Course: Atributy tabulky Course

CourseResult	typ	velikost	PK	null	def.	popis
IdCourse	nvarchar	20	1	0		ID kurzu (Course.Id)
IdUser	nvarchar	20	1	0		ID studenta (User.Id)
AcademicYear	nvarchar	9	1	0		Akademický rok ve tvaru „YYYY/YYYY“
Points	float		0	0		Získaný počet bodů
Grade	nvarchar	20	0	1		Výsledná známka

Tabulka A.CourseResult: Atributy tabulky CourseResult

InspectLog	typ	velikost	PK	null	def.	popis
Id	bigint		1	0		ID přístupu do systému (inkrementační)
IP	nvarchar	20	0	0		IP adresa přístupu
Port	int		0	0	-1	
Browser	nvarchar	max	0	1		Prohlížeč uživatele
Path	nvarchar	max				Cesta v souborovém systému serveru
SubSystem	nvarchar	max	0	1		Subsystém systému (Barborka)
Method	nvarchar	max	0	1		(Barborka)
IdCourse	nvarchar	20	0	1		ID kurzu (Course.Id)
IdUser	int		0	0	-1	ID uživatele (User.Id)
UserGroup	nvarchar	20	0	1		role
Duration	bigint		0	0	-1	Doba zobrazení stránky
DurationPow	bigint		0	0	-1	Kvadrát doby zobrazení stránky
Start_Ticks	bigint		0	1		Datum a čas začátku připojení (Time.Ticks)
End_Ticks	bigint		0	1		Datum a čas konce připojení (Time.Ticks)

Tabulka A.InspectLog: Atributy tabulky InspectLog

Question	typ	velikost	PK	null	def.	popis
Id	nvarchar	20	1	0		ID otázky (z původní databáze)
Type	nvarchar	20	0	1		Typ otázky – A-B-C-D, textová, … (Barborka)
Name	nvarchar	max	0	1		Název otázky
Time	int		0	1		Předpokládaný čas přečtení otázky
Text	nvarchar	max	0	1		Text otázky
AuthorId	int		0	0	-1	ID autora otázky (User.Id)
CourseId	nvarchar	20	0	1		ID kurzu (Course.Id)
IdCathegory	int		0	1		Id kategorie otázky (Cathegory.Id)

Tabulka A.Question: Atributy tabulky Question

SchoolClass	typ	velikost	PK	null	def.	popis
IdStudent	int		1	0		ID studenta (User.Id)
IdTeacher	int		1	0		ID učitele (User.Id)
Type	int		1	0		Typ třídy (1=přednáška, 2=cvičení)
AcademicYear	nvarchar	9	1	0		Akademický rok ve tvaru „YYYY/YYYY“
Id	int		0	0	-1	ID třídy (z původní databáze)
IdCourse	int		0	0	-1	ID kurzu (Course.Id)

Tabulka A.SchoolClass: Atributy tabulky SchoolClass

Test	typ	velikost	PK	null	def.	popis
Id	int		1	0		ID testu (z původní databáze)
IdCourse	nvarchar	20	0	1		ID kurzu (Course.Id)
IdUser	int		0	0	-1	ID uživatele (User.Id)
Points_max	float		0	0		Maximální počet bodů
Question_num	int		0	0	-1	Počet otázek
Time	int		0	0	-1	Čas vypracovávání testu
Type	int		0	0	-1	0=elektronický, 1=papírový

Tabulka A.Test: Atributy tabulky Test

TestGroup	typ	velikost	PK	null	def.	popis
Id	int		1	0		ID termínu (z původní databáze)
IdTest	nvarchar	20	0	0	-1	ID testu (Test.Id)
Type	int		0	1		Typ aktivity
Capacity	int		0	1		Kapacita termínu
Number	int		0	0		Pořadí termínu (inkrementační v rámci stejného IdTest)
Students	float		0	1		Počet přihlášených studentů
Start_Ticks	bigint		0	1		Čas začátku termínu (Time.Ticks)
End_Ticks	bigint		0	1		Čas konce termínu (Time.Ticks)

Tabulka A.TestGroup: Atributy tabulky TestGroup

TestResultQuestion	typ	velikost	PK	null	def.	popis
Id	nvarchar	20	1	0		Kombinace IdTest a TestNum
IdQuestion	nvarchar	20	1	0	-1	ID otázky (Question.Id)
IdTest	int		0	0	-1	ID testu (Test.Id)
IdUser	int		1	0		ID uživatele (User.Id)
TestNum	int		0	0	-1	Varianta testu
Answer	nvarchar	max	0	1		Odpověď (pokud je textová)
Duration	int		0	1		Doba zobrazení otázky
Training	bool		0	0	true	Identifikátor cvičný/hlavní test (cvičný test = true)
Points	float		0	0	-1	Získané body
Points01	float		0	0	-1	Získané body normované na interval <0;1> (pokud je možné získat záporné body, minimum je převedeno na nulu a maximum na jedna)
Points01P	float		0	0	-1	Nezáporná část bodů normovaná na interval <0;1> (jedná se o pozitivní zisk, všechny případné záporné body jsou převedeny na nulu, maximum je jedna)
PointsPow	float		0	0	-1	Kvadrát Points01
Points_max	float		0	0	-1	Maximum bodů k získání
Points_min	float		0	0	-1	Minimum bodů k získání
Time_Ticks	bigint		0	1		Začátek řešení otázky (Time.Ticks)

Tabulka A.TestResultQuestion: Atributy tabulky TestResultQuestion

TestResultAnswer	typ	velikost	PK	null	def.	popis
Id	nvarchar	20	1	0		Kombinace IdTest a TestNum
IdQuestion	nvarchar	20	1	0		ID otázky (Question.Id)
IdVariant	nvarchar	20	1	0		ID odpovědi (Variant.Id)
IdTest	int		0	0	-1	ID testu (Test.Id)
IdUser	int		1	0		ID uživatele (User.Id)
TestNum	int		0	0	-1	Varianta testu
Correct	bit		0	0	0	Správnost zodpovězení odpovědi
Time_Ticks	bigint		0	1		Čas zodpovězení odpovědi (Time.Ticks)
Training	bool		0	0	true	Identifikátor cvičný/hlavní test (cvičný test = true)
Clicks	int		0	1		Počet kliků na odpověď

Tabulka A.TestResultAnswer: Atributy tabulky TestResultAnswer

User	typ	velikost	PK	null	def.	popis
Id	int		1	0		ID uživatele (z původní databáze)
Name	vnarchar	200	0	0	-1	Jméno a příjmení

Tabulka A.User: Atributy tabulky User

TestUser	typ	velikost	PK	null	def.	popis
IdTest	int		1	0		ID test (Test.Id)
TestNum	int		1	0		Varianta testu
IdUser	int		1	0		ID studenta (User.Id)
IdTestGroup	int		0	0	-1	ID termínu (TestGroup.Id)
Points	float		0	0	-1	Získané body
Duration	bigint		0	0	-1	Doba trvání testu
PointsPow	float		0	0	-1	
DurationPow	bigint		0	0	-1	
Start_Ticks	bigint		0	1		Začátek řešení testu (Time.Ticks)
End_Ticks	bigint		0	1		Konec řešení testu (Time.Ticks)
Training	bool		0	0	true	Identifikátor cvičný/hlavní test (cvičný test = true)
Try	int		0	0		Pokus o absolvování testu

Tabulka A.TestUser: Atributy tabulkyTestUser

Time	typ	velikost	PK	null	def.	popis
Ticks	bigint		1	0		Milisekundy od 1.1.1901
DayOfMonth	int		0	0	-1	Den v měsíci
Month	int		0	0	-1	Měsíc
DayOfWeek	int		0	0	-1	Den v týdnu
AcademicYear	nvarchar	10	0	0	-1	Akademický rok
Hour	int		0	0	-1	Hodina
Minute	int		0	0	-1	Minuta
Second	int		0	0	-1	Sekunda
Year	int		0	0	-1	Rok

Tabulka A.Time: Atributy tabulky Time

Příloha B: *Uživatelská příručka*

Na https://elogika.vsb.cz je vytvořen uživatelský účet **vsbtest** s heslem **697dd04d61**. V levém menu je odkaz na stránku s data miningovou aplikací.

Volba parametrů a zobrazení statistické analýzy

Obrázek 15 *Volba parametrů statistické analýzy*

Uživatel může zvolit několik druhů statistické analýzy, podle toho, co chce analyzovat:

- **Analýza testů** - analyzuje výsledky z vybraných testů
- **Analýza kategorií** - analyzuje výsledky otázek podle zařazení do kategorií
- **Analýza otázek** - analyzuje výsledky otázek
- **Analýza odpovědí** - analyzuje výsledky odpovědí

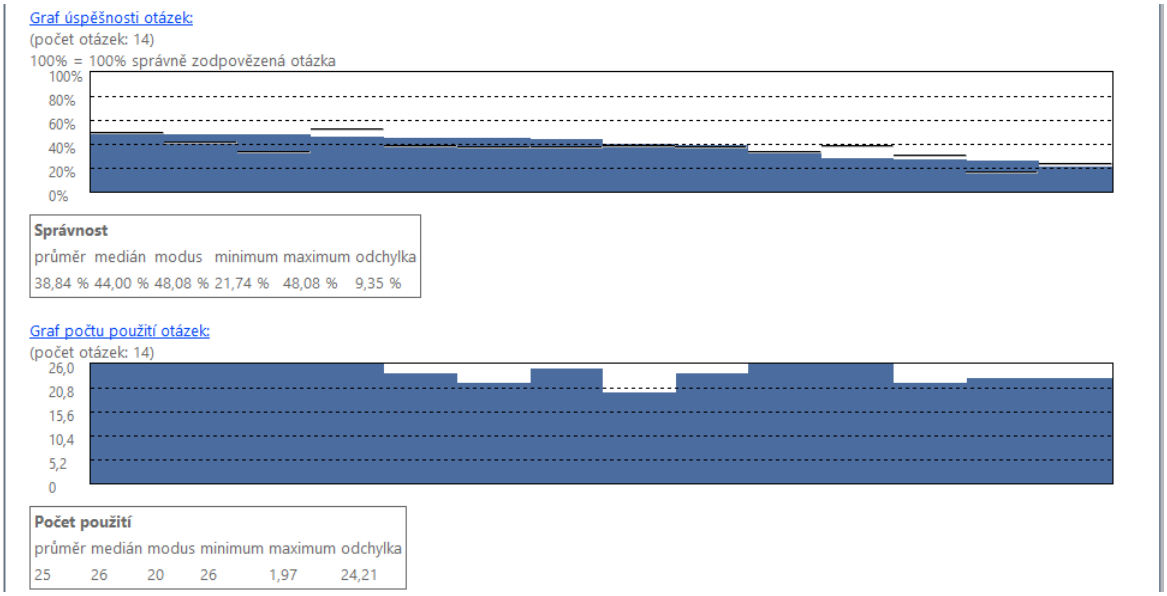
Uživatel má možnost zvolit několik parametrů:

- **Předmět** - Na začátku analýzy je potřeba zvolit předmět, jehož výsledky chceme analyzovat.
- **Započítané roky** - Zde má uživatel možnost zvolit filtr pro akademické roky, jejichž výsledky se mají použít pro analýzu. Navíc je zde možné zvolit jinou množinu roků pro výpočet celkového průměru.

- **Forma studia** - Tento filtr slouží k použití výsledků pouze určité formy studia (prezenční/kombinovaní). Je použit i při výpočtu celkového průměru. Není dostupný pro data mining ze systému Barborka.
- **Tutor** - Pomocí tohoto filtru můžeme analyzovat výsledky studentů konkrétních tutorů. Není dostupný pro data mining ze systému Barborka.
- **Výsledek** - Tento filtr slouží k použití výsledků pouze určité typu. Můžeme analyzovat všechny výsledky, cvičné výsledky, nebo hlavní výsledky (relevantní pro hodnocení kurzu). Je použit i při výpočtu celkového průměru.
- **Test** - Tento filtr umožňuje analyzovat pouze konkrétní testy.

Specifika analýzy otázek a odpovědí:

- **Hraniční úspěšnost** - Pomocí těchto parametrů uživatel zvolí interval, ve kterém se musí nacházet procentuální správnost zodpovězení otázky (odpovědi) aby byla zobrazena. Základní nastavení je <0;100>, tedy vyhledání všech záznamů. Uživatel by se měl zaměřit na otázky (odpovědi), které mají nízkou (<0;20>) nebo naopak vysokou (<80;100>) správnost zodpovězení.
- **Hraniční počet použití** - Pomocí těchto parametrů uživatel zvolí interval, který určuje, kolikrát byla otázka (odpověď) použita aby mohla být zobrazena. Základní nastavení je <0;--->, tedy vyhledání všech záznamů. Přesnější jsou samozřejmě informace o otázkách (odpovědích), které byly použity víckrát.
- **Kategorie** - Tento filtr umožňuje analyzovat otázky a odpovědi otázek z konkrétních kategorií.

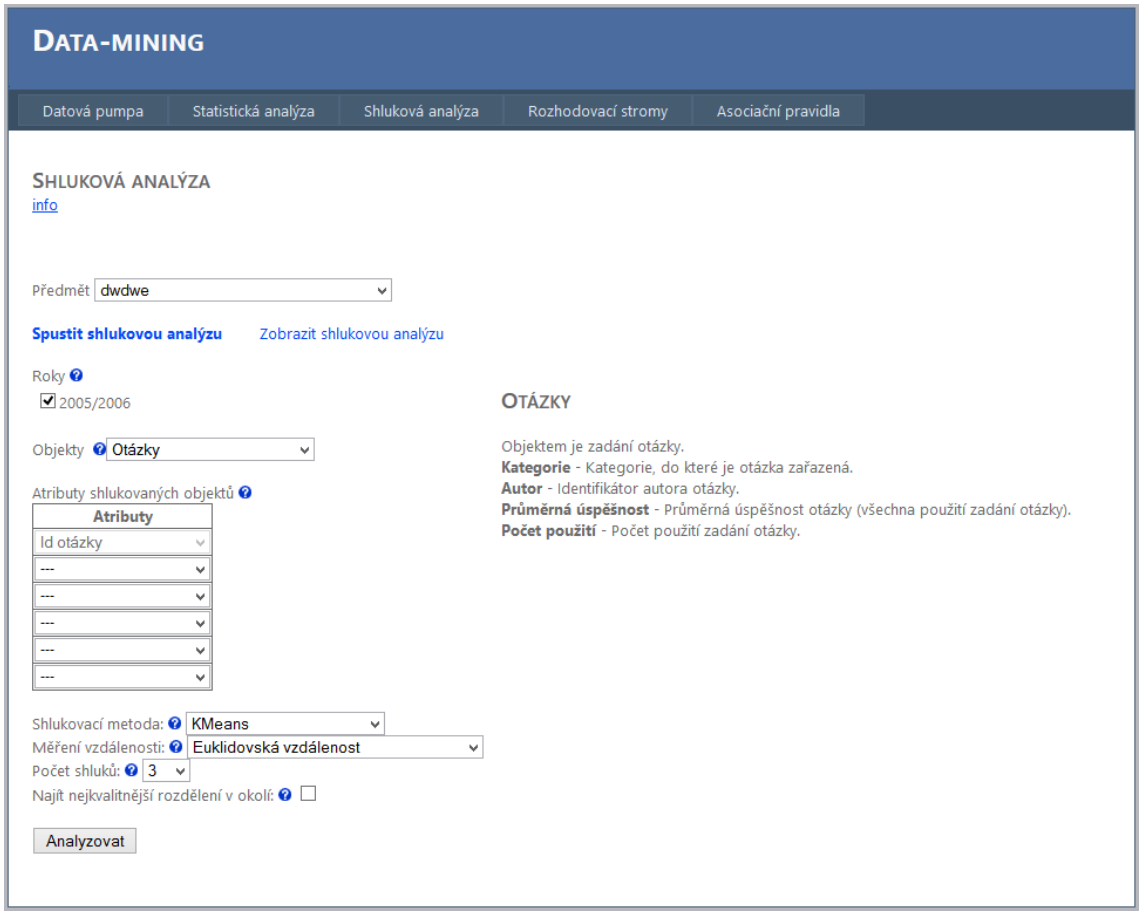


Obrázek 16 *Zobrazení statistické analýzy*

Po dokončení analýzy se zobrazí dva grafy (graf úspěšnosti a graf počtu použití). První

graf obsahuje průměrnou úspěšnost (modrý sloupec) a celkovou průměrnou úspěšnost (černá čára). Druhý graf obsahuje počet výsledků. U každého grafu je také tabulka se základními statistickými hodnotami, jako průměr, medián, modus, minimum, maximum a odchylka.

Volba parametrů shlukové analýzy



Obrázek 17 *Volba parametrů shlukové analýzy*

Uživatel musí zvolit analyzovaný předmět a akademické roky, ve kterých jej chce analyzovat. Následně musí zvolit objekt, který ho zajímá, tedy jakou část databáze chce analyzovat a atributy, které chce použít.

Uživatel si může vybrat z několika dostupných shlukovacích algoritmů:

- **K-Means** - Algoritmus pro shlukování numerických atributů. Rozdělí objekty do k shluků tak, aby byl každý objekt v tom shluku, od jehož středu je nejméně vzdálený.
- **Hierarchické shlukování** - Algoritmus pro shlukování numerických atributů. Rozdělí objekty do shluků pomocí algoritmu single linkage. Jedná se agromerativní shlukování, které postupně spojuje nejbližší objekty, až dosáhne hledaného počtu shluků.

- **QuickROCK** - Algoritmus pro shlukování kategoriální atributů. Spojuje objekty, které jsou si podobnější, než zadaná prahová hodnota (threshold).

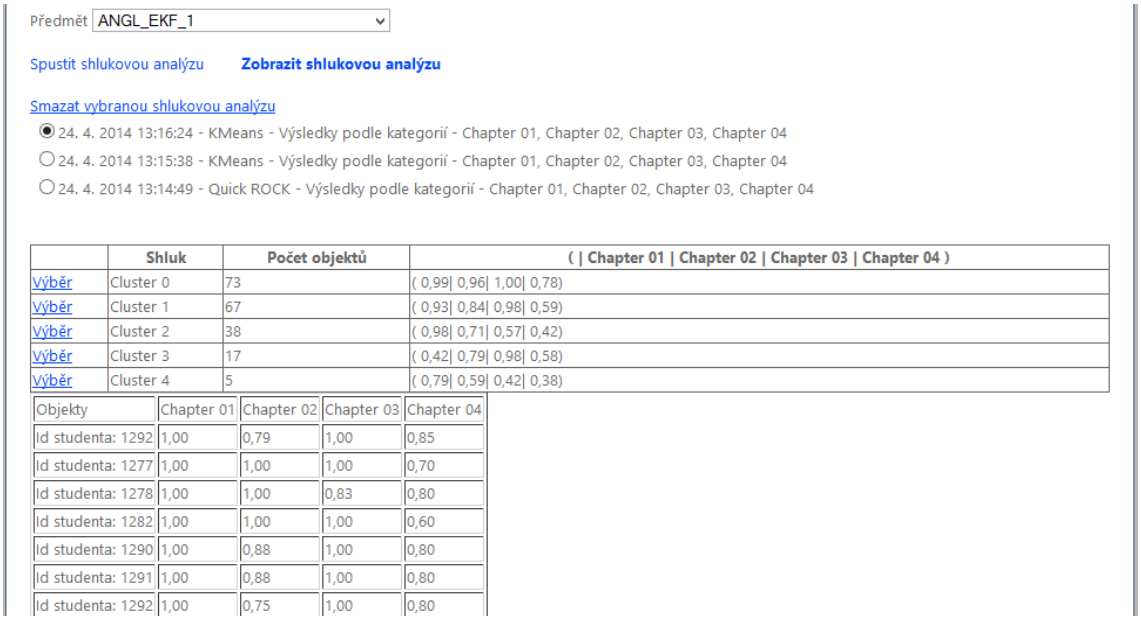
U shlukovacích algoritmů pro numerická data (k-means a hierarchické shlukování) je možné zvolit:

- **Způsob měření vzdálenosti** - Uživatel si může vybrat mezi euklidovskou a manhattanskou vzdáleností. Kromě toho je k dispozici i normovaná varianta, která zajistí, aby měly všechny atributy stejný význam.
- **Počet shluků** - Uživatel musí vybrat počet shluků, které mají vzniknout.
- **Hledání nejkvalitnějších rozdělení** - Při zvolení této varianty se provede shluková analýza několikrát (podle zvolené velikosti okolí) a vybere se takové rozdělení shluků, které má nejlepší kvalitu podle zadaného hodnocení kvality.

U shlukovacího algoritmu pro kategoriální data (quickROCK) je možné zvolit:

- **Threshold** - Threshold je prahová hodnota, která udává, kdy jsou si dva objekty ještě podobné. Threshold může v případě algoritmu quickROCK nabývat pouze několika hodnot (v závislosti na počtu atributů vybraných pro shlukování). Hodnota "0" značí, že dva objekty se mohou lišit ve všech atributech a přesto jsou si podobné. Hodnota "1" značí, že dva objekty musí mít všechny atributy stejné, aby si byly podobné.

Zobrazení shlukové analýzy

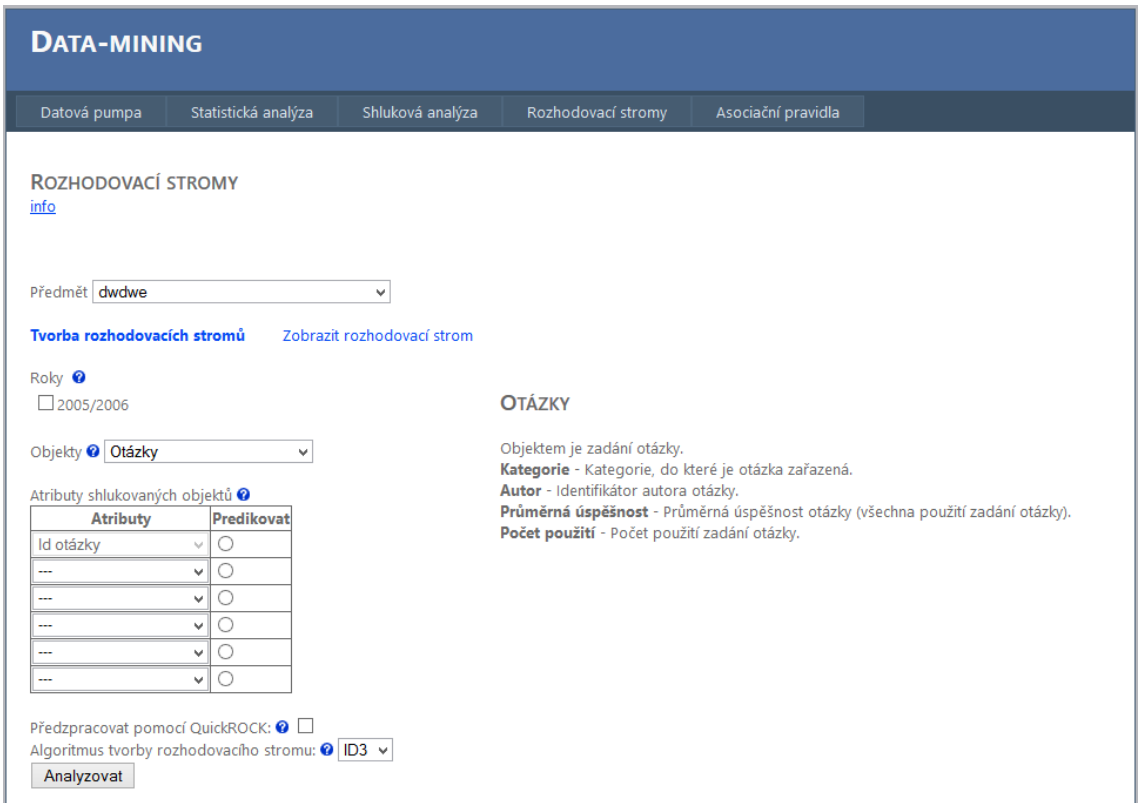


Obrázek 18 *Zobrazení shlukové analýzy*

Po vytvoření shlukové analýzy je uživateli zobrazena stránka s již vytvořenými shlukovými analýzami. Tuto stránku je možné zobrazit i kliknutím na odkaz „Zobrazit shlukové analýzy“. Na této stránce je seznam všech shlukových analýz vybraného kurzu. Po výběru

konkrétní analýzy je tato zobrazena ve formě tabulky. Každý řádek tabulky představuje jeden shluk. Výběrem konkrétního shluku se uživateli zobrazí tabulka s obsahem tohoto shluku. Kliknutím na odkaz „Smazat vybranou shlukovou analýzu“ se analýza smaže.

Volba parametrů rozhodovacího stromu



Obrázek 19 *Volba parametrů rozhodovacího stromu*

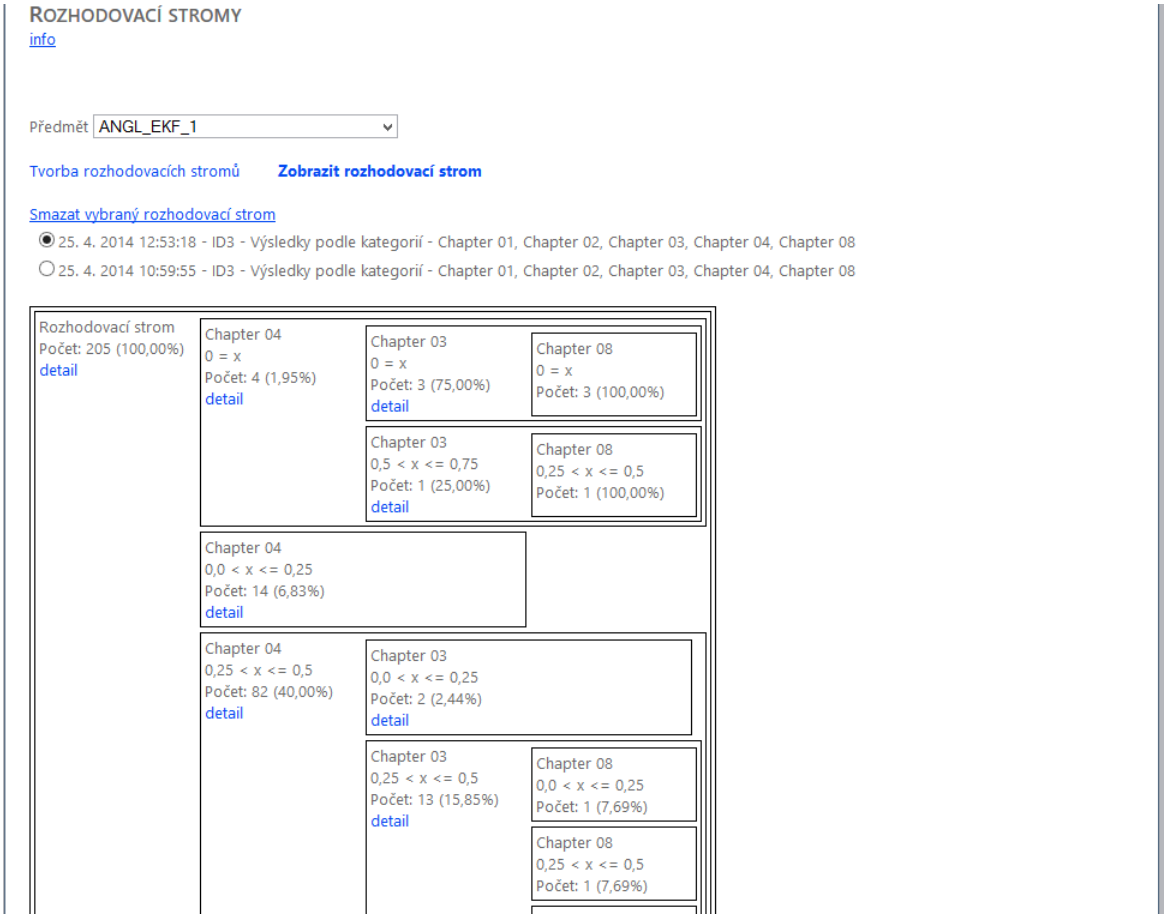
Uživatel musí zvolit analyzovaný předmět a akademické roky, ve kterých jej chce analyzovat. Následně musí zvolit objekt, který ho zajímá, tedy jakou část databáze chce analyzovat a atributy, které chce použít.

Uživatel může vybrat volbu **Předzpracovat pomocí QuickROCK**. Toto nastavení umožňuje první spojit všechny identické objekty a až následně spustit algoritmus tvorby rozhodovacích stromů. Toto nastavení je výhodné zejména v případě, že se v databázi nachází velké množství záznamů a vybrané parametry mají malý počet hodnot.

Zobrazení rozhodovacích stromů

Po vytvoření rozhodovacího stromu je uživateli zobrazena stránka s již vytvořenými stromy. Tuto stránku je možné zobrazit i kliknutím na odkaz „Zobrazit rozhodovací strom“. Na této stránce je seznam všech rozhodovacích stromů vybraného kurzu. Po výběru konkrétního stromu je tento zobrazen ve formě tabulky. Kliknutím na odkaz „Smazat vybraný rozhodovací

strom“ se strom smaže. Uživatel má u každé větve stromu možnost skrýt ji kliknutím na odkaz „detail“, aby mohl získat lepší představu a celkový pohled.



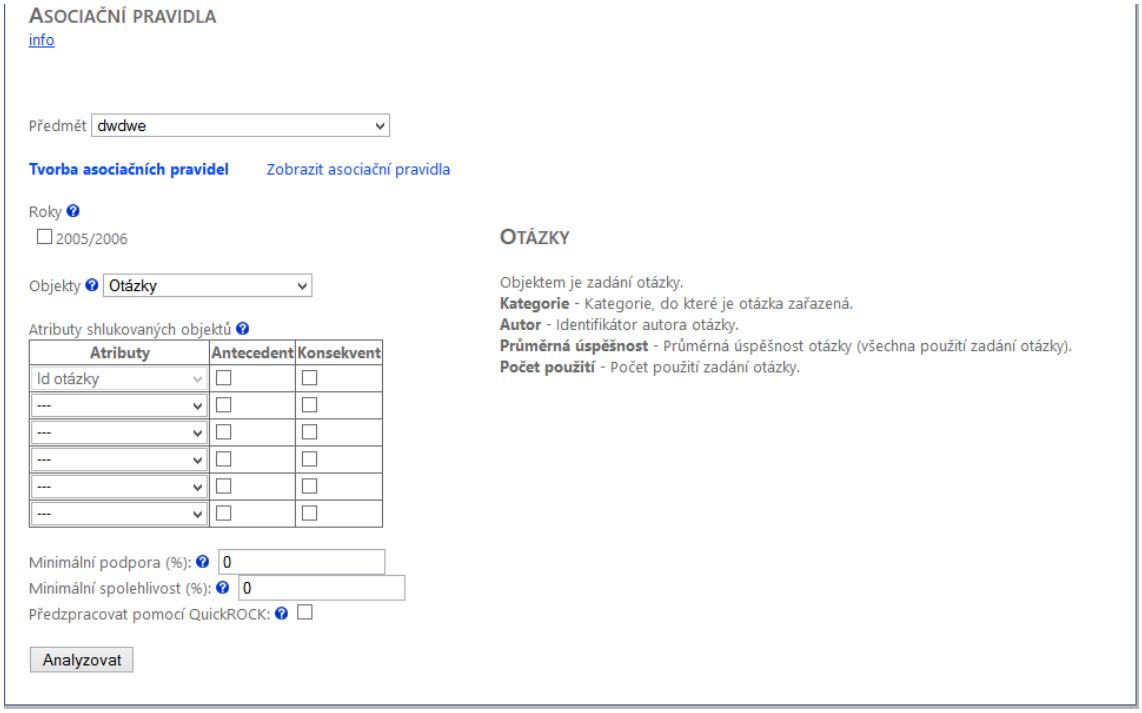
Obrázek 20 *Zobrazení rozhodovacího stromu*

Volba parametrů asociačních pravidel

Uživatel musí zvolit analyzovaný předmět a akademické roky, ve kterých jej chce analyzovat. Následně musí zvolit objekt, který ho zajímá, tedy jakou část databáze chce analyzovat a atributy, které chce použít.

U zvolených atributů je nutné zvolit jestli se jedná o antecedent (příčinu) nebo o konsekvent (následek).

- **Antecedent** - Levá strana asociačního pravidla. Tyto atributy mohou být použity jako "příčina".
- **Konsekvent** - Pravá strana asociačního pravidla. Tyto atributy mohou být použity jako "následek".



Obrázek 21 *Volba parametrů asociačních pravidel*

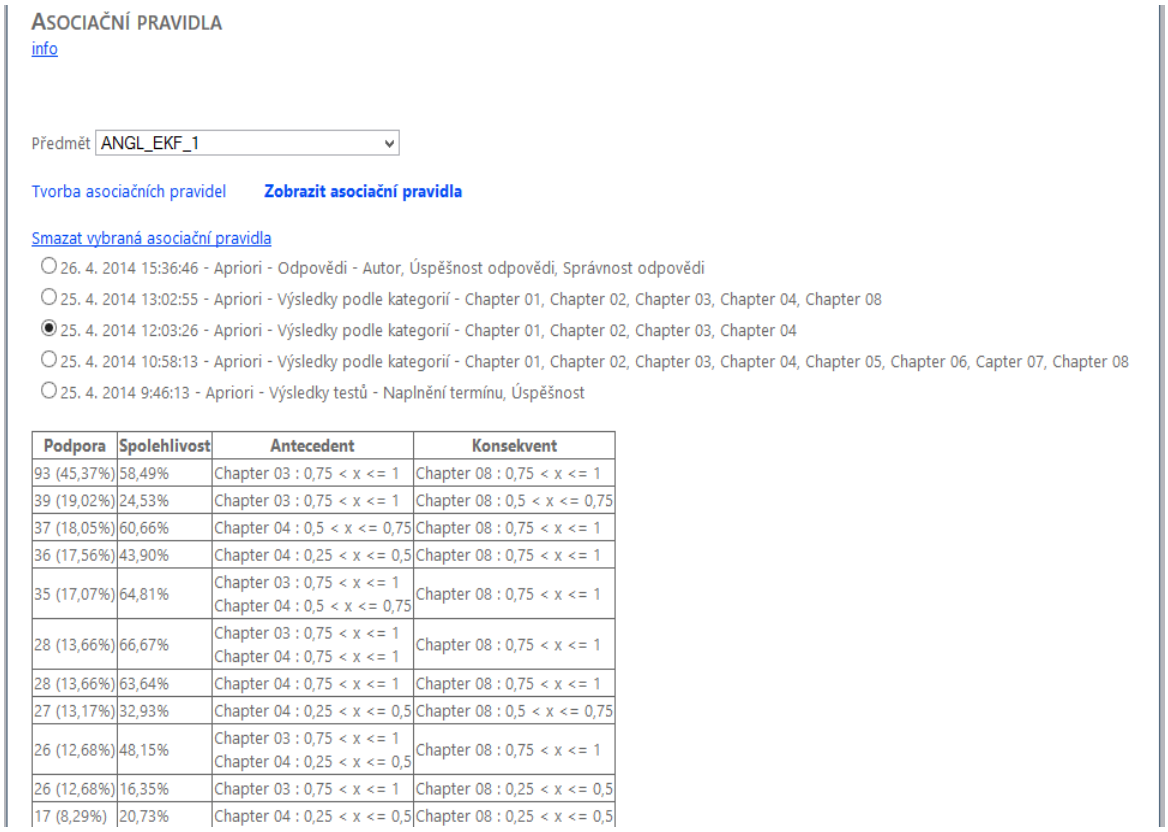
Uživatel musí vybrat parametry asociačních pravidel:

- **Minimální podpora** - Minimální podpora p značí, že kombinace hodnot (např. (A,B)) bude započítána v případě, že se vyskytuje alespoň v p procentech záznamů. Jedná se o pravděpodobnost výskytu této kombinace.
- **Minimální spolehlivost** - Minimální spolehlivost s značí, že z kombinace hodnot (např. (A,B)) vznikne pravidlo (např. (A→ B)) pokud je poměr mezi záznamy s touto kombinací hodnot (A,B) a všemi záznamy s antecedentem (A) alespoň s. Jedná se o podmíněnou pravděpodobnost.

Uživatel může vybrat volbu **Předzpracovat pomocí QuickROCK**. Toto nastavení umožňuje první spojit všechny identické objekty a až následně spustit algoritmus tvorby rozhodovacích stromů. Toto nastavení je výhodné zejména v případě, že se v databázi nachází velké množství záznamů a vybrané parametry mají malý počet hodnot.

Zobrazení asociačních pravidel

Po vytvoření asociačních pravidel je uživateli zobrazena stránka s již vytvořenými pravidly. Tuto stránku je možné zobrazit i kliknutím na odkaz „Zobrazit asociační pravidla“. Na této stránce je seznam všech asociačních pravidel vybraného kurzu. Po výběru konkrétních pravidel jsou tato zobrazena ve formě tabulky. Kliknutím na odkaz „Smazat vybraná asociační pravidla“ se pravidla smažou.



Podpora	Spolehlivost	Antecedent	Konsekvent
93 (45,37%)	58,49%	Chapter 03 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
39 (19,02%)	24,53%	Chapter 03 : 0,75 < x <= 1	Chapter 08 : 0,5 < x <= 0,75
37 (18,05%)	60,66%	Chapter 04 : 0,5 < x <= 0,75	Chapter 08 : 0,75 < x <= 1
36 (17,56%)	43,90%	Chapter 04 : 0,25 < x <= 0,5	Chapter 08 : 0,75 < x <= 1
35 (17,07%)	64,81%	Chapter 03 : 0,75 < x <= 1 Chapter 04 : 0,5 < x <= 0,75	Chapter 08 : 0,75 < x <= 1
28 (13,66%)	66,67%	Chapter 03 : 0,75 < x <= 1 Chapter 04 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
28 (13,66%)	63,64%	Chapter 04 : 0,75 < x <= 1	Chapter 08 : 0,75 < x <= 1
27 (13,17%)	32,93%	Chapter 04 : 0,25 < x <= 0,5	Chapter 08 : 0,5 < x <= 0,75
26 (12,68%)	48,15%	Chapter 03 : 0,75 < x <= 1 Chapter 04 : 0,25 < x <= 0,5	Chapter 08 : 0,75 < x <= 1
26 (12,68%)	16,35%	Chapter 03 : 0,75 < x <= 1	Chapter 08 : 0,25 < x <= 0,5
17 (8,29%)	20,73%	Chapter 04 : 0,25 < x <= 0,5	Chapter 08 : 0,25 < x <= 0,5

Obrázek 22 *Zobrazení asociačních pravidel*